*Proceedings of the 2025 Winter Simulation Conference*
*E. Azar, A. Djanatliev, A. Harper, C. Kogler, V. Ramamohan, A. Anagnostou, and S. J. E. Taylor, eds.*

# SIMULATING THE DYNAMIC INTERACTION BETWEEN FLEET PERFORMANCE AND MAINTENANCE PROCESSES BASED ON REMAINING USEFUL LIFE

Christoph Werner[1]

[1]Minitab / Simul8, 77 Renfrew Street, Glasgow, UK

## ABSTRACT

Fleet planning is often challenging. Especially, the dynamic interaction with maintenance entails various uncertainties. Predicting the arrivals into maintenance processes requires an understanding of fleet performance over time while, in turn, delays of repairs severely impact fleet performance and deterioration.

This feedback loop has been neglected so far, which is why we present a novel framework using a 'rolling window' machine learning model to predict the inputs into a discrete event simulation (DES) of repair activities based on remaining useful life (RUL). Our 'fleet tracker' then uses the DES outputs to simulate fleet performance together with environmental and mission-based factors which form the inputs for predicting RUL. Finally, explainable ML helps decision-makers construct relevant 'what if' scenarios. As a motivating example, we consider helicopters in search and rescue missions and their maintenance. As a key result, we compare two scenarios of repair turnaround times and their impact on RUL decline.

## 1 INTRODUCTION

Fleet management and planning is a complex challenge for many organizations. In particular, understanding the dynamic interaction between fleet performance and the required maintenance, repair and overhaul (MRO) processes to ensure a safe and acceptable level of service and operation poses an important, yet so far neglected, problem. As examples we can think of organizations managing fleets of assets, such as trucks, ships, aircraft or helicopters, that are being used in supply chains and ambulance services, or even deployed in missions and training, for instance in search and rescue (SAR) (coastguards and mountain rescue), or also the military. In all of these examples, it is important to consider and understand (1) the impact that fleet performance has on managing resources and capacities in an MRO facility (e.g., due to a sudden surge in demand for repairs or needing to prioritize mission critical assets), but also (2) the impact that, in turn, constraints and delays (bottlenecks) in the MRO process can have on fleet performance (e.g., assets in a fleet deteriorating quicker than expected due to absorbing work of assets delayed in repair). This interaction becomes an even more complex challenge when considering that we are often dealing with globally deployed fleets operating in a wide variety of environments. In some cases, these are organizations producing components, such as engines, that are vital for assets in a fleet to function and for which they provide technical assistance and maintenance for several customers.

In this paper, we present a novel approach to tackle and simulate this dynamic interaction stemming from the previous two points as a feedback loop. Our first contribution is the technical implementation. We present a new way to determine the arrival process of a discrete event simulation (DES) based on a common concept in predictive maintenance, that of remaining-useful-life (RUL). For that, we integrate a 'rolling window' machine learning (ML) model, predicting RUL based on a 'fleet tracker', with a DES of a maintenance and repair process. The 'fleet tracker' is a stochastic simulation with agent-based elements that captures accumulated operating hours of each asset in a fleet together with environmental and mission-specific factors and preferences that might impact their RUL (e.g., how long was an asset operating in adverse weather or difficult mission types in remote locations?). However, it is not simulating operating hours in isolation. Instead, these depend on the assets available in a fleet each day, which itself depends on the maintenance process DES and possible delays and bottlenecks therein. Thereby the inputs into all models constitute a loop to capture the dynamics of the aforementioned two points as shown in Figure 1.

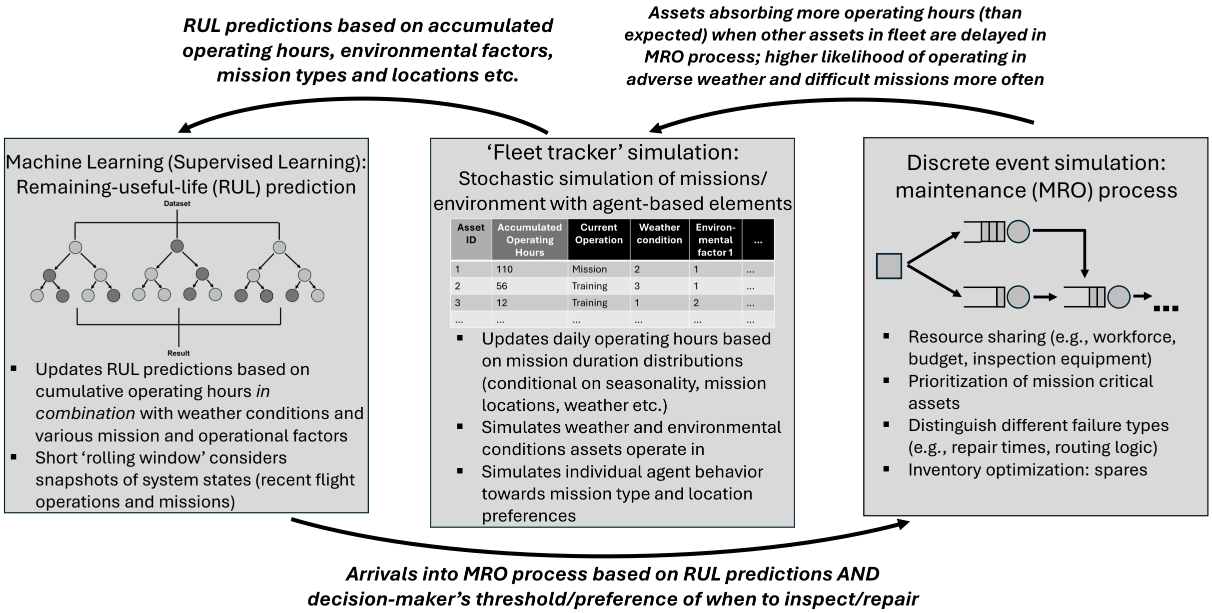


Figure 1: Schematic overview of integrating supervised learning, a 'fleet tracker' simulation and DES.

Given that RUL predictions can change rapidly through a sudden increase in accumulated operating hours once a fleet size is diminished due to delayed repairs, especially in combination with adverse weather and difficult mission types, our second contribution considers the decision-makers. Their responsibility is to decide when assets come in for inspection and repair. Although we might determine a fixed threshold for RUL below which assets should come into maintenance, for many fleets it might be beneficial to base such decisions on the current state of the MRO facility. As such, one might bring in assets earlier than their RUL suggests if there are currently no bottlenecks. This balances the current workload with the expected future one. Therefore, we let decision-makers explore the RUL predictors via explainable ML and inform 'what if' scenarios through them. For example, they can test what happens if a long period of difficult mission types or extreme weather was to occur while fleet size is reduced? This factors in any uncertainty around RUL predictions, especially for combinations of predictors for which only scarce data exist. DePater et al. (2022) already highlighted the importance of considering imperfect RUL predictions.

The remainder of this paper is as follows. Section 2 briefly reviews the literature on integrating ML methods with DES, focusing on simulation input modeling, and ML in predictive maintenance. In section 3, we discuss where our approach sits within fleet management and predictive maintenance research before, in section 4, we present our framework together with results from a case study based on a SAR helicopter fleet. Finally, section 5 concludes the paper by outlining the main findings and future enhancements.

## 2 LITERATURE REVIEW

We reviewed the literature with regards to two objectives. First, we show why and how ML algorithms have been integrated with DES models, especially for input modeling. Second, we present the ML methods commonly used in predictive maintenance more generally.

### 2.1 Machine Learning for Simulation Input Modeling

Practitioners and researchers integrate ML with simulations for various reasons, including learning (complex) conditional logic for prioritization and routing of entities, (Bergmann et al. 2017), automating results interpretation (Mladenić et al. 1994), and meta-modeling (for optimization) (Fishwick 1989). However, the reason that is of main interest to us is that of enhancing distributional assumptions in input modeling. For that, we considered primarily Supervised Learning (SL) methods. They improve their prediction accuracy through training from labeled examples in large datasets (see e.g. Kubat (2017)). Thus, we briefly summarize the use of Unsupervised Learning (USL), which aims at finding or clusters (structures) in unlabeled data, and Reinforcement Learning (RL), concerned with agents learning best actions to take in an environment through rewards, in DES models as follows. The former can improve the understanding of input data. However, USL algorithms are typically not integrated directly into DES models, i.e., they do not update their inputs during nor between simulation runs. Instead, they help exploring data before conceptualizing and constructing simulations. Elbattah et al. (2018) used USL to cluster patient data when simulating a hip fracture care pathway. RL is integrated with DES to use simulations as 'test beds' (learning environments) that allow agents to learn optimal strategies in current system states. Eriksson et al. (2022) used RL to identify optimal schedules in a robot cell manufacturing DES.

More aligned with our interests, SL-based classification and regression methods have been used previously in input modeling, and especially whenever inputs depend on multiple attributes (of entities) or the simulation state, ML methods can be more flexible (than fitting distributions). For instance, in a job shop DES, (Frye et al. 2019) predicted workpieces' processing times via SL based on their individual current conditions. Similarly, Jungmann et al. (2022) used decision trees (DTs) and random forests (RFs) to set processing time inputs of crane operations in a construction DES. Further, Azab et al. (2021) compared various SL methods (boosted DTs and neural networks (NNs)) to predict machine breakdown durations in a flow-shop DES. All references considered that workpieces and machines are distinguished by current conditions and individual attributes, so that SL provided more accurate inputs than fitted distributions. In healthcare, Fairley et al. (2019) trained gradient boosting methods on patient data to set treatment and recovery durations in a post-anesthesia care DES while Ortiz-Barrios et al. (2024) used RFs to predict the probability of patients' respiratory diseases worsening (resulting in longer bed occupancy) in an A&E DES. Further, Chang and Chang (2018) estimated treatment times with NNs in a dental care simulation while Kashani et al. (2024) and Olave-Rojas and Nickel (2021) both used RFs to predict length-of-stay (LoS) durations of hospital patients and estimate ambulance travel times accordingly.

While the previous examples mainly predicted processing, LoS and breakdown durations, the next two references are more aligned with us by considering arrival processes and frequencies. Glowacka et al. (2009) used association rules to capture 'no shows' of patients in a healthcare DES informing the arrival process, and Theeuwes et al. (2021) trained DTs on past ambulance dispatching decisions for determining arrivals. Nevertheless, SL for arrival processes is much less common so far.

### 2.2 Machine Learning for Predictive Maintenance

Similar to the previous approaches predicting simulation inputs, which might depend on several attributes, conditions and system states, SL-based prediction impacts also the area of predictive maintenance more generally. Here we are often interested in predicting RUL and Zhang et al. (2018) used a long short-term memory (LSTM) NN to forecast the degradation of a turbofan engine (with data from an open-source dataset by NASA and containing sensor readings). Their regression model estimated the remaining cycles of the engine before maintenance will be required. Using the same dataset, Mathew et al. (2017) compared various ML methods and concluded that tree-based approaches (RFs and gradient-based boosting) lead to the most accurate predictions. Indeed, SL-methods are now commonly applied to predict RUL and overviews of methods are provided in Ferreira and Gonçalves (2022) and Berghout and Benbouzid (2022).

## 3 BACKGROUND: FLEET MANAGEMENT AND PREDICTIVE MAINTENANCE

Next, we outline the concepts that our framework builds upon in more detail. This embeds it in the research areas of fleet management and maintenance strategies while highlighting the contributions to these areas.

### 3.1 Fleet Management

Given that the arrivals into our MRO simulation are driven by the overall behavior of a fleet, we should briefly define a fleet and its main characteristics. According to Petchrompo and Parlikad (2019), a fleet constitutes a group of assets that are similar regarding their technical features while they often share common maintenance facilities, resources and interventions. Fleet management is then the act of overseeing and organizing such fleets to ensure safe and efficient operations while fulfilling operational demand. The point of sharing resources (spare parts, workforce, tools, budget) and facilities for maintenance and repair is particularly relevant for us as our DES serves exactly this purpose. We support the planning and optimization of resources together with the overall planning of the maintenance facility needed to provide timely repairs and minimize unplanned downtime that might lead to knock-on effects on the remaining fleet in operation. This helps decision-makers to better understand the interaction within a fleet regarding its performance and degradation which is often misunderstood as consisting of independent assets. In the literature, this idea of adapting fleets to current conditions is known as *dynamic fleet management* (del Castillo et al. 2023) – an area where DES can make important contributions. So far, most models supporting dynamic fleet management rely purely on optimization approaches, such as mixed-integer linear and non-linear programing (Petchrompo and Parlikad (2019) discuss examples). Thereby they omit possible dynamic considerations and uncertainties about the current state of a maintenance facility (and its capacity, bottlenecks, resource availability etc.). This can lead to the aforementioned feedback loop of active assets operating more than planned and degrading unexpectedly whenever other assets are delayed in repair. Further, DES can include details such as the impact of prioritizing mission critical assets on possible repair delays.

### 3.2 Predictive Maintenance and Remaining Useful Life

The importance of selecting a suitable maintenance strategy is well understood by decision-makers in fleet management. It increases reliability and reduces costs while avoiding unplanned downtime (Pintelon et al. 2006). The main challenge concerns the timing of a maintenance operation, i.e., when to inspect and possibly repair an asset. Generally, we distinguish three approaches: *reactive*, *preventive*, and *predictive* maintenance (Swanson 2001). For us, predictive maintenance is of interest. It estimates the optimal time to inspect and repair based on the predicted time of failure. The inherent challenge concerns, of course, building an accurate model for detecting anomalies and predicting failures. A key concept, often being the predicted output, is RUL. It provides the time between the current condition and failure for a given asset through its deterioration over time (Si et al. 2011). RUL is often provided as a time unit (e.g., days, hours), however, other measures are possible, such as the number of cycles. We estimate it in one of three ways depending on the available data. *Survival* models require failure times from similar assets, *similarity* models need run-to-failure data from similar assets, and *degradation* models use condition indicators to predict future changes and when a chosen safety threshold is crossed. We mostly align with the last approach.

Although improvements in digital technologies (sensors) and more data-driven maintenance approaches let fleet managers nowadays respond to issues more easily, some challenges persist, such as sudden changes in requirements and operating in new environments. These exemplify the possible uncertainty around RUL predictions and examples are helicopters doing missions in new (possibly extreme) weather conditions or accumulating unusually many flight hours in remote locations. In our framework, we include this uncertainty and predict RUL therefore differently. Instead of considering input vectors of regular sensor updates, such as the aforementioned NASA turbofan data, our RUL predictions are based on various aggregated features derived from a rolling window of recent flight cycles for each engine, capturing environmental and operational conditions per flight in addition to categorical flight mission and location profiles. Thus, we consider aggregated tabular data from multiple possible influencing factors of RUL where each row represents a snapshot of the system state, i.e., a daily summary of recent flight metrics. We do not necessarily assume strong time dependence of time-windows (or snapshots) as some factors can be assumed independent. This is different from common approaches to anomaly detection that solely focus on degradation data, thus omitting the impact of environmental factors on future predictions. It allows us to simulate the likely future conditions of engines by considering the upcoming flight environment (or system states more generally). Hence, we identify patterns of short sequences of events prior to sharp drops in RUL. This avoids the need to consider data from an asset's whole trajectory. Further, our predictions come as distributions which reflects maintenance decisions for fleets more realistically.

## 4 OUR FRAMEWORK

After having discussed our position within the research areas of fleet management and predictive maintenance strategies, we now present our framework in detail. Our motivating example considers fleet planning for SAR missions. More specifically, we considered a producer of engine parts for helicopters deployed in missions, such as the UK Coastguard Agency, for which they offer technical assistance and MRO support.

### 4.1 'Fleet Tracker' Simulation: Stochastic Modeling of Mission Dispatch and Helicopter Allocation

As shown in Figure 1, the 'fleet tracker' simulation sits at the center of our framework. To capture fleet performance over time, it simulates the daily scheduling of missions and training flights for a fleet of four helicopters, incorporating the associated randomness and operational constraints. It is a hybrid between a stochastic model by considering mission demand, durations, type and environmental factors probabilistically, and an agent-based model by incorporating the preferences and specifications of individual helicopters that impact their likelihood of flying a given mission.

Our goal is to simulate how mission demand, environmental conditions, and mission characteristics interact to affect helicopter utilization, thereby highlighting periods of increased strain on workload, and ultimately their RUL. Based on a global simulation clock, which also tracks time in the DES, we sample the number of missions for each day following a non-homogeneous Poisson process. Its intensity function varies with seasonality and weather conditions. A SARIMA (Seasonal Auto Regressive Integrated Moving Average) model provides the expected number of monthly missions based on historical SAR helicopter mission data of the UK Maritime & Coastguard Agency (MCA) (including trends, seasonality and autocorrelations)(see MCA (2025)). Here, we see i.a. that July and August are regular annual peaks. Depending on monthly patterns, we compute the likely number of daily missions which are adjusted by a weather multiplier. Different categorical weather types (i.a., clear, cloudy, rainy, stormy, snowy, foggy) are chosen with monthly season-dependent probabilities to modify the mission rate. For instance, storms boost mission numbers and are more likely during winter months.

Next, we define mission types and locations. The former distinguishes rescue (known and unknown), medical assistance, support, pre-arranged transfer and search missions. Mission type probabilities depend on seasonality and weather whereas in MCA (2025) rescue made up most mission types (48%). The locations are also distinguished categorically. While the MCA considers coast, land and maritime, we extended these to distinguish mountain, mountain-remote and urban areas in the land category. The location type probabilities depend on mission type, e.g., rescue and support missions are more likely in maritime environments (MCA 2025). Both mission type and location impact the mission duration. Most missions last between 1 and 9 hours, but some extend over multiple days. We capture this with several log-normal distributions with parameters conditional on mission types and locations. Some exhibit a corresponding long right tail to specify the uncertainty of remote locations and rescue and search missions.

Finally, we assign daily missions to available helicopters. Of course, the input of available helicopters is an output from the DES model. First, we schedule high-priority tasks (like rescue missions). Typically, helicopters with fewer flight hours since last maintenance are assigned first. However, we also include individual agent behavior, such as some helicopters being preferred for specific mission types or durations. For instance, missions at sea more suitable with certain helicopter sizes. Therefore, we assume a fleet with two larger and two smaller helicopters. On days when the number of missions exceeds helicopters available, we either assign multiple missions to a single helicopter, or reschedule lower priority missions. Whenever daily missions are less than helicopters available, we check for training flights. Although we could apply a fixed schedule for these, we consider a probabilistic approach, capturing the chance of cancellations.

The 'fleet tracker' was implemented in Python and records accumulated flight hours per helicopter together with the conditions of each flight. That way, it informs the RUL predictions presented next. For example, capturing that a helicopter has flown several missions over sea recently, especially in stormy conditions, can impact its vibration and lead to a sharp drop in RUL. At a more granular level, we can include simulated vibration levels conditional on mission types and environmental factors if data from monitoring are available. In a similar way we can distinguish the categorical weather conditions more granularly by sampling conditional temperatures, wind speeds and humidity levels. Figure 2 shows a screenshot.

```
Day 11:
  Helicopter 1 → ON mission: Rescue-unknown | Cloudy | Mountain | 9h
  Helicopter 2 → ON mission: Training | Rain | Remote-Mountain | 8h
  Helicopter 3 → ON mission: Rescue-known | Cloudy | Urban | 2h
  Helicopter 4 → ON mission: Rescue-known | Cloudy | Sea | 7h
Day 12:
  Helicopter 1 → ON mission: Rescue-known | Rain | Remote-Mountain | 8h
  Helicopter 2 → UNAVAILABLE
    → Replacement Helicopter 3 takes over: Training | Clear | Remote-Mountain | 4h
  Helicopter 4 → ON mission: Search Only | Clear | Mountain | 2h
```

Figure 2: The 'fleet tracker' simulates and logs detailed mission data per helicopter per day, including mission type, weather, location, duration, and role (primary or replacement).

The above idea of flying many short missions reflects SAR operations realistically. However, in other contexts, including military and large-scale emergencies like forest fires, a mission profile likely differs by requiring an intense period of flying for a longer period, e.g., several weeks, typically followed by much less flying afterwards (for a while). The 'fleet tracker' simulation can be adapted to that.

### 4.2 Machine Learning-Based Regression: Predicting RUL of SAR Helicopters

As mentioned in section 2.2, RUL prediction is becoming more data-driven (rather than based on physical models) and several ML models have been proposed for that purpose. We use DTs and RFs as they handle linear and non-linear relationships while also capturing complex interactions between features in high-dimensional data. Indeed, we consider them a suitable model choice for including aggregated features for environmental and operational conditions that are easier to train than most deep learning alternatives. However, to account for the aims set out in section 3.2, our model differs from more common RF approaches in two important ways:

1. We include the uncertainty around RUL predictions,
2. We identify patterns of recent system states (snapshots of last flights) that lead to a sharp drop in RUL (short 'rolling window').

DTs and RFs are some of the most common SL algorithms and, as outlined in section 2.1, they learn from labeled datasets. After training on example data, DTs construct classification or regression trees, either predicting discrete or continuous variables. For RUL we consider the latter. Internal nodes in a DT, i.e., not leaves, have input labels and are considered decision rules. Leaf nodes on the other hand contain predicted output values that can be traced back to the DT's root node. During training, DTs are split repeatedly to minimize prediction errors according to a chosen loss function (e.g., MSE). Given DTs' propensity to overfitting, i.e. when predictive performance is good in training but poor with new data, RFs have been introduced as extension (Breiman 2001). They are an ensemble method and construct multiple DTs, using a bootstrapped subset of examples and inputs to reduce the risk of overfitting.

Although RFs in regression problems typically output the average of all predictions, we use the percentiles of all trees (the tree-wise prediction spread) to create an uncertainty aware RF with estimated confidence intervals. Our 'rolling window' approach to estimate RUL based on high-level operational data, i.e., the temporally aggregated flight, weather and mission features, informed by the 'fleet tracker' simulation, can be summarized as follows. First, we decide on the number of time windows, i.e., last flights to consider (in our case 5). Then, we aggregate the features from those flights, i.e., get averages for continuous variables, such as wind speeds, in addition to frequencies and fractions for categorical variables, for instance, the different mission types and locations. This data is tabular and resembles columns in the 'fleet tracker', so each row provides an input vector. Hence, the feature aggregation over a short 'rolling window' provides the RF model with a memory of the most recent flights (system state snapshots). However, we should clarify that we do not assume strong time dependence but only leverage local patterns, as ideally, we then identify short-term patterns that occur right before a decline in RUL. For instance, we might observe a sharp drop after flying multiple missions over sea in cold temperatures or after flying several rescue missions. At this point, we can also add the simulated maintenance history, provided by the DES, to enhance RUL predictions. E.g., it might be of interest if a part had been repaired or replaced in the last MRO visit. The RF with short rolling windows was constructed in Python using the scikit-learn library (Pedregosa et al. 2011). Here, we also supplemented our training examples through a synthetic dataset due to a lack of real flight data to test our approach. Figure 3 shows an example screenshot of RUL predictions.

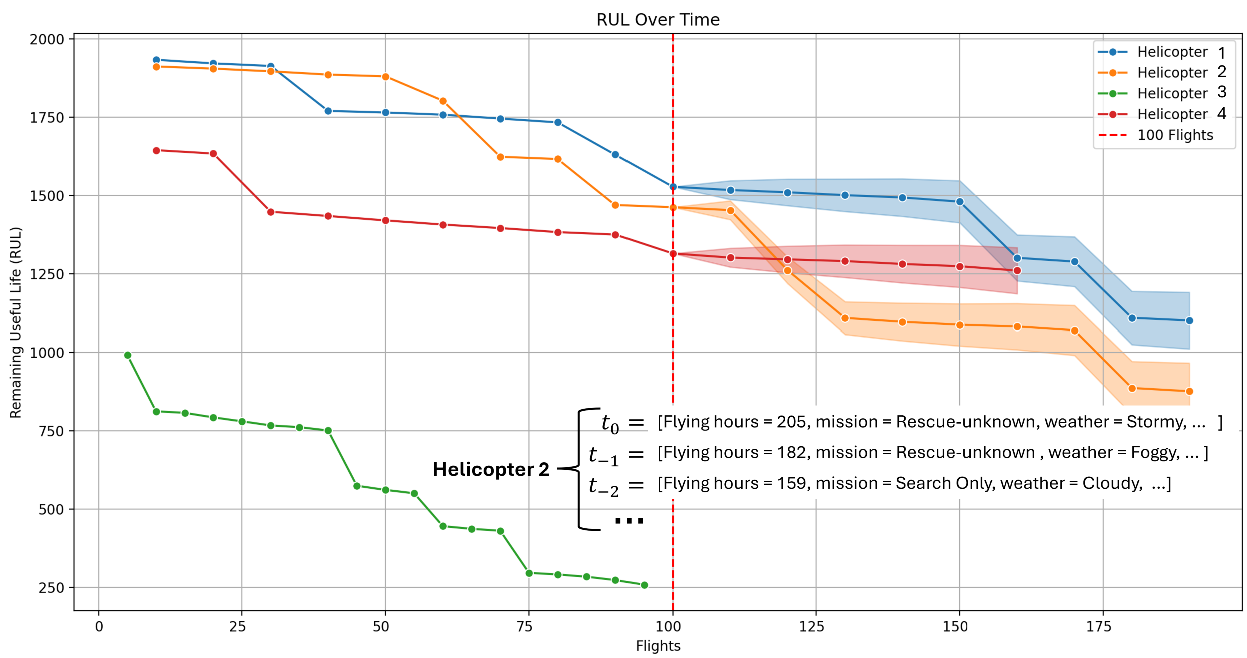


Figure 3: Example predictions of helicopters after 100 fights in combination with cumulative flying hours, environmental and mission factors).

This plot shows the RF predictions of the individual RUL changes per helicopter. As such, Helicopter 2 is predicted to see a sharp drop in RUL as it has recently accumulated many flying hours in difficult missions (rescue in remote locations) and adverse weather conditions at $t_0 = 100$, $t_{-1} = 90$ and $t_{-2} = 80$ flights. This was caused (1) by Helicopter 2 being a large model and the other corresponding large one, Helicopter 3, not being available due to reaching its end of RUL and going to MRO (Helicopter 2 absorbed most longer and more remote missions during that time) and (2) due to seasonality with worse weather conditions more likely in the last time-windows. As this pattern likely continues for the next flights RUL might decrease further. Thus, it is not only the uncertainty about flight hours accumulating with a reduced fleet, but also about most likely mission and weather related factors. The benefits of our approach are the consideration of multi-flight historical data, capturing (short) temporal patterns influencing engine degradation, and the inclusion of mission profile effects. In the next section, we discuss the impact that the time-to-repair (TTR) of a helicopter, i.e., the duration of flying with an incomplete fleet, has on the decrease in RUL predictions.

### 4.3 Discrete Event Simulation: Modeling the Maintenance and Repair Process

DES modeling is commonly used to better understand and analyze a wide variety of processes and for a general introduction, we refer to Robinson (2014) and Law and Kelton (2000). Thus, it also gained popularity in MRO planning and Alrabghi and Tiwari (2015) present an overview. It provides maintenance managers and engineers with a way for risk-free experimentation, scenario testing and optimization of important levers (or decision variables) they can influence, such as the number of available resources, and even process flow considerations. In line with the discussion in section 3.1, we want to show the impact that sharing workforce, machinery and spare parts has on the repair turnaround time in order to return engine parts to the helicopters while also including additional important details, such as prioritization based on mission criticality. We used Simul8 for building the DES and a screenshot is shown in Figure 4.

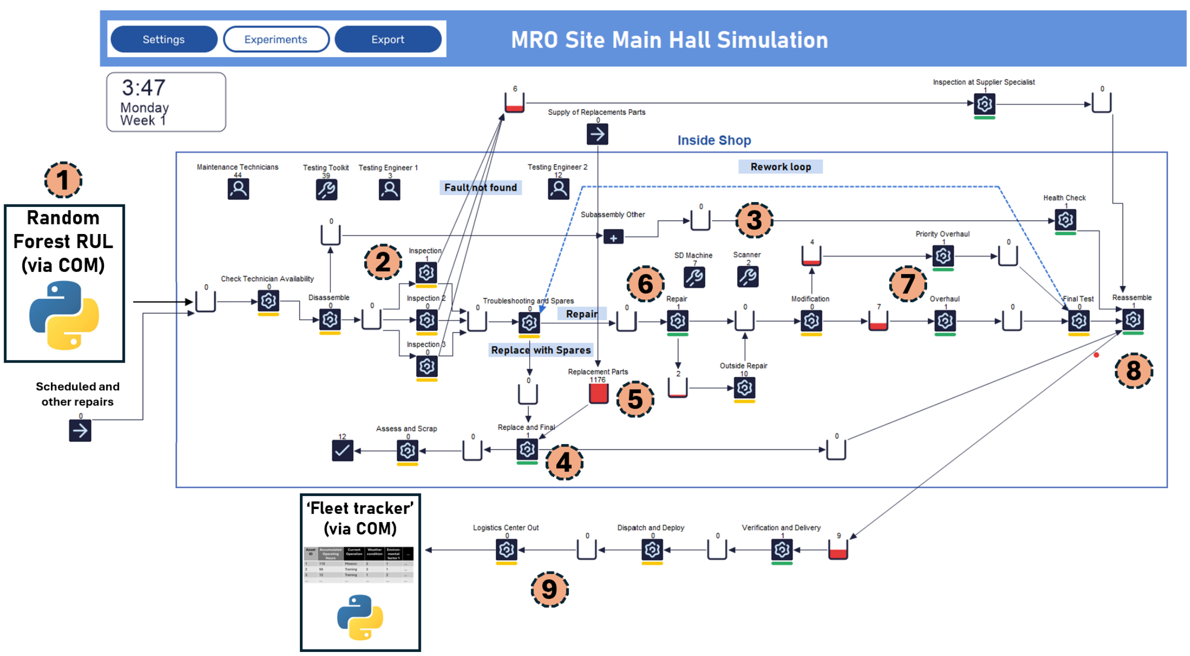


Figure 4: Main part of maintenance and repair (MRO) process simulation (screenshot Simul8).

The process starts with incoming engine parts (see point 1 in Figure 4). The arrival frequency depends on their RUL and the decision of when to repair (see section 4.2). In addition, we have scheduled (preventive maintenance) and unscheduled arrivals from other fleets for which the MRO facility is also responsible. As mentioned in the introduction, this is common with companies maintaining components for assets.

Any incoming part requires two maintenance technicians. The first makes an initial inspection to assess the fault type and severity. This decides if repairs go ahead (point 2). Some are dismissed at this stage due to fault severity while others are sent to the original supplier or a more specialized technical crew, for instance, if the fault cannot be identified. The second technician stays with each repair job until final testing at the end of the process. Note, from here onwards, we consider that each engine has been disassembled, and repairs relate to individual components. Thus, any component not needing repair is routed aside for later reassembly into a whole engine again (point 3). Before repairable parts go to the main repair activity, we check if we can simply replace the faulty component subject to replacement part availability (point 4). Replacement parts are replenished regularly but can deplete if demand becomes too high (point 5). When replaced, the turnaround for engines back to the fleet is typically quicker as the actual repair can be done later while no helicopter is 'on ground'. If not possible, several specialized tools are needed at the main repair activity (point 6) depending on failure type, summarized here as SD machines. Here, we can prioritize repair of jobs if they are mission critical. After modification we might prioritize mission critical parts again for overhaul (point 7). This means that technicians and machinery are used there first, and ongoing work might be interrupted if necessary. Next, components are tested for a final time whereas there is a chance of failure and joining the rework loop back to troubleshooting and inspection. After passing final testing, repaired components are reassembled into the complete engine part and leave the facility (point 8). They are then cleared for delivery and dispatched back to the fleet (point 9).

While we could add more detail of MRO processes, this is out of scope here. We just note that despite most individual processing times being reasonably assumed to follow log-normal distributions with low standard deviations, hence, not causing unexpected delays by themselves, several factors can introduce high uncertainties for predicting the overall TTR or turnaround time for engines returning to fleets. These include (1) sharing resources and spares together with possible unavailabilities, (2) mission critical assets taking priority, and (3) different routing options, e.g., to a specialist crew or for rework. This also explains why solely considering mean-time-to-repair (MTTR) is often an oversimplified assumption. The main exceptions for uncertainty stemming from distributional assumptions relate to the inspection and repair activities. These can vary greatly due to the different possible causes of engine faults.

Regarding the main results in interaction with the 'fleet tracker' simulation and individual helicopter's RUL predictions, we compare two possible system states of the MRO DES, called scenario 1 and 2 respectively. Note, these are just two examples with respect to how busy the MRO process is due to repairs from other fleets (either scheduled or unscheduled). They were chosen to show the impact of repair delays on RUL and vice versa after running trial runs of the simulations in this framework. Scenario 1 reflects a 'less busy' scenario with an exponential inter-arrival rate of other arrivals into MRO of 36 hours. Scenario 2 reflects a 'somewhat busy' scenario with other arrivals coming in every 12 hours (exponential). In scenario 1, the repairable part from the first helicopter engine reaching the end of its RUL had an average time-in-system of 271.36 hours, 95% CI [269.65,273.07] (measured until Reassembly) while the average sum of all waiting times was 29.59 hours, 95% CI [33.91,25.27]. During this time, the average RUL for the three remaining helicopters dropped by 84.68 flying hours. When running the 'fleet tracker' simulation for the same length of time with a complete fleet, the average RUL reduction is only 56.81 flying hours. This is caused by the slope of each RUL estimate being corrected downwards once the first helicopter becomes inactive. In scenario 2, the average time-in-system was 391.36 hours, 95% CI [381.98,400.75] with a total average wait time of 97.55 hours, 95% CI [87.65,107.45]. Further, time-in-system had a long right tail with the $95^{th}$ quantile at 579.35 hours signifying a risk as the average RUL of the three other helicopters decreased already by 154.77 flying hours. This behavior indicates a lagged phenomenon we can plan with when investigating the impact that specific features have on RUL as explored next. That way, the results inform multiple decisions. First, they can minimize the impact that MRO has on a fleet by timing it with the current state of the facility. This avoids getting caught in bottlenecks and through 'what if' scenarios, we can assess how well an incomplete fleet copes with such scenarios and recovers from them, especially with the risk of operational readiness collapsing within this feedback loop. This goes beyond simply preparing for a busier summer season, but instead allows for informed decisions when facing the uncertainty of challenging missions and locations together with unexpected adverse weather.

### 4.4 Explainable ML: Making 'What If' Scenario Development Accessible for Decision-Makers

A common challenge with ML-based predictions concerns the understanding, transparency and interpretation of outputs. Indeed, most ML models are known as 'black boxes' (Castelvecchi 2016). This also raises an important concern when integrating simulation models with ML, which is why we include this step in our modeling process. It allows decision-makers to gain a better understanding of the drivers of sharp drops in RUL and by that the aforementioned dynamic interaction between fleet performance and maintenance. Explainable ML has emerged as an umbrella term for techniques that aid gaining a better insight into the factors that predictions are based upon (see Belle and Papantonis (2021) for an overview). In our case decision-makers need to understand when fleet performance becomes critical for maintenance planning and vice versa. We achieve this through integrating a common explainable ML method, Shapley Additive Explanations (SHAP). It is based on Shapley values from cooperative games in game theory (Shapley 1953), whereas here Shapley values calculate the weighted marginal contribution of variables to the final prediction result of the RF model. We refer to Štrumbelj and Kononenko (2014) for details. The method is model-agnostic, suitable for a small to moderate number of features, and it overcomes issues with ordering.

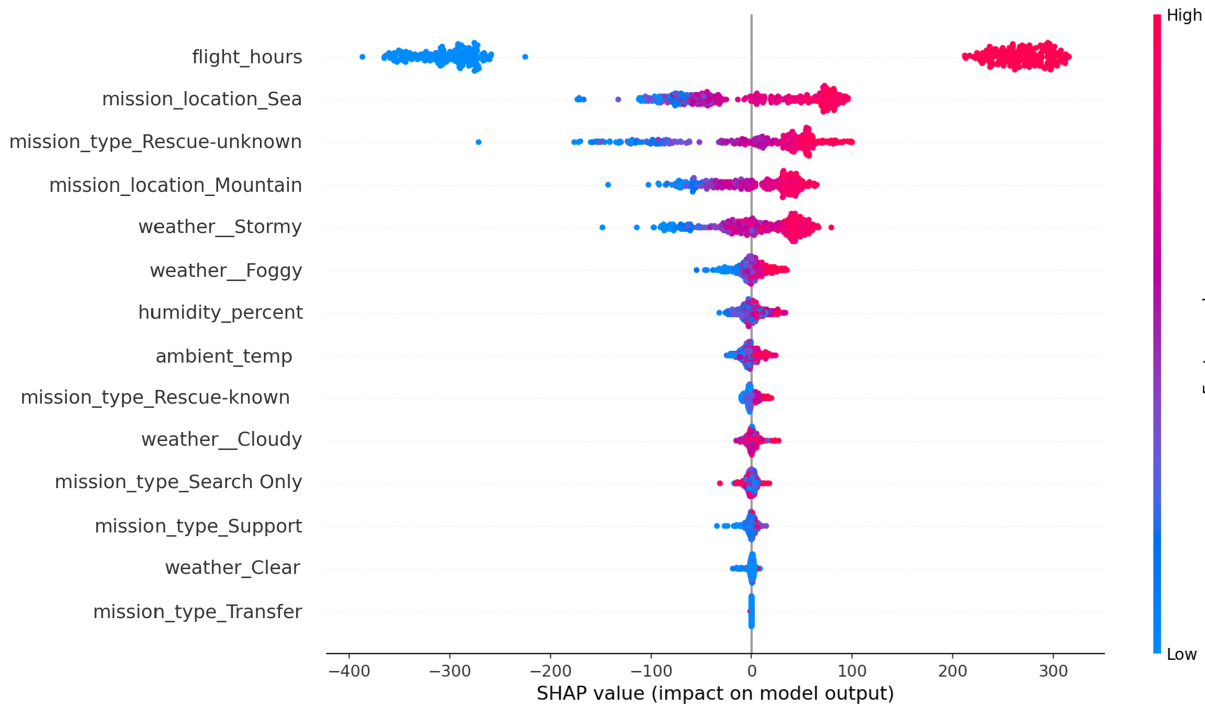


Figure 5: SHAP values evaluating the contributions of individual features on RUL decrease).

In Figure 5 (built in Python's SHAP package), we see that especially mission locations *Sea* and *Mountain* together with mission type *Rescue unknown* and stormy and foggy weather have an impact on our RUL predictions. All of these come (unsurprisingly) after the actual accumulated flight hours. The clusters of high values (red) in contrast to spread out low values (blue) indicate that these are not just individual outliers causing occasional increase in SHAP values. Further, we see that the SHAP values separate the important and less important features well which confirms the outputs of the 'fleet tracker' simulation in the sense that it distinguishes relevant features for RUL. Next, we can use the SHAP values to assess the reliability of our simulation results. In particular, whenever we only have scarce data on the most important features, we should consider ways to counteract this lack of data, such as including expert judgments to assess their likelihood. For example, with a lack of data on remote mission probabilities, we could elicit this information to improve the 'fleet tracker' simulation. Werner (2023) discussed simulation input modeling in the absence of relevant historical data. Finally, these results (and their correlations) are useful for developing informative 'what if' scenarios that stress test the current MRO set up, attuned to coping with the current expected drops in RUL. For instance, we might simulate other mission type probabilities combined with adverse weather to specifically include and mimic real-world conditions in SAR missions.

## 5 CONCLUSIONS

In this paper, we have addressed the dynamic interaction between fleet performance and MRO processes through a novel simulation framework. It considers the feedback loop that exists between these given that delays and bottlenecks in repairs can put stress on fleet performance which, in turn, might cause a rapid decline in RUL and, hence, more arrivals into MRO again. Our simulation framework, integrating an RF model with DES through a 'fleet tracker' provides a more holistic view of this problem. In addition, it includes important factors at the individual level of the fleet assets that can cause a sudden decline in RUL.

In future, we would like to validate the framework in real-world applications and further develop it. First, although the idea of a 'rolling window' for RFs provides realistic results of short-term sequences leading to drops in RUL, we want to explore more advanced methods. In that context, LSTM considers time series data specifically and can overcome the less flexible fixed time windows of feature vectors in RFs. Second, other models are better suited to capture uncertainty around RUL predictions by treating model parameters probabilistically and predicting quantiles from trees, such as Bayesian approaches for RFs or also Quantile Regression Forests. Next, the 'fleet tracker' and DES model can be enhanced to account for important additional details. The first with respect to agent-based details of helicopter pilots and their behavior and the latter regarding technical assistance and maintenance for global fleets, including the impact of multiple maintenance facilities and rules regarding first and second line support.

Finally, we want to enhance our framework with a simulation-optimization approach considering the states of all models together. This allows us to benchmark it against existing fleet management optimization and simulation approaches. Further, by integrating interactive methods for exploring scenarios of declining RUL, such as serious games based on explainable ML, we can test the robustness of its results while making them more informative and accessible for decision-makers, especially when basing 'what if' testing on live (or near live) data, e.g., on the current state of the fleet or current weather, for operational use.

## REFERENCES


Alrabghi, A., and A. Tiwari. 2015. "State of the Art in Simulation-Based Optimisation for Maintenance Systems". *Computers & Industrial Engineering* 82:167–182.

Azab, E., M. Nafea, L. A. Shihata, and M. Mashaly. 2021. "A Machine-Learning-Assisted Simulation Approach for Incorporating Predictive Maintenance in Dynamic Flow-Shop Scheduling". *Applied Sciences* 11(24):11725.

Belle, V., and I. Papantonis. 2021. "Principles and Practice of Explainable Machine Learning". *Frontiers in Big Data* 4:688969.

Berghout, T., and M. Benbouzid. 2022. "A Systematic Guide for Predicting Remaining Useful Life with Machine Learning". *Electronics* 11(7):1125.

Bergmann, S., N. Feldkamp, and S. Strassburger. 2017. "Emulation of Control Strategies through Machine Learning in Manufacturing Simulations". *Journal of Simulation* 11(1):38–50.

Breiman, L. 2001. "Random Forests". *Machine Learning* 45:5–32.

Castelvecchi, D. 2016. "Can We Open the Black Box of AI?". *Nature* 538(7623):20–23.

Chang, W. J., and Y. H. Chang. 2018. "Design of a Patient-Centered Appointment Scheduling with Artificial Neural Network and Discrete Event Simulation". *Journal of Service Science and Management* 11(1):71–82.

del Castillo, A. C., J. A. Marcos, and A. K. Parlikad. 2023. "Dynamic Fleet Maintenance Management Model Applied to Rolling Stock". *Reliability Engineering & System Safety* 240:109607.

DePater, I., A. Reijns, and M. Mitici. 2022. "Alarm-Based Predictive Maintenance Scheduling for Aircraft Engines with Imperfect Remaining Useful Life Prognostics". *Reliability Engineering & System Safety* 221:108341.

Elbattah, M., O. Molloy, and B. P. Zeigler. 2018. "Designing Care Pathways Using Simulation Modeling and Machine Learning". In *2018 Winter Simulation Conference (WSC)*, 1452–1463 https://doi.org/10.1109/WSC.2018.8632360.

Eriksson, K., S. Ramasamy, X. Zhang, Z. Wang, and F. Danielsson. 2022. "Conceptual Framework of Scheduling Applying Discrete Event Simulation as an Environment for Deep Reinforcement Learning". *Procedia CIRP* 107:955–960.

Fairley, M., D. Scheinker, and M. L. Brandeau. 2019. “Improving the Efficiency of the Operating Room Environment with an Optimization and Machine Learning Model”. *Health Care Management Science* 22:756–767.

Ferreira, C., and G. Gonçalves. 2022. “Remaining Useful Life Prediction and Challenges: A Literature Review on the Use of Machine Learning Methods”. *Journal of Manufacturing Systems* 63:550–562.

Fishwick, P. A. 1989. “Neural Network Models in Simulation: A Comparison with Traditional Modeling Approaches”. In *1989 Winter Simulation Conference (WSC)*, 702–709 https://doi.org/10.1145/76738.76828.

Frye, M., D. Gyulai, J. Bergmann, and R. H. Schmitt. 2019. “Adaptive Scheduling through Machine Learning-Based Process Parameter Prediction”. *MM Science Journal* 63:3060–3066.

Glowacka, K. J., R. M. Henry, and J. H. May. 2009. “A Hybrid Data Mining/Simulation Approach for Modelling Outpatient No-Shows in Clinic Scheduling”. *Journal of the Operational Research Society* 60(8):1056–1068.

Jungmann, M., L. Ungureanu, T. Hartmann, H. Posada, and R. Chacon. 2022. “Real-Time Activity Duration Extraction of Crane Works for Data-Driven Discrete Event Simulation”. In *2022 Winter Simulation Conference (WSC)*, 2365–2376 https://doi.org/10.1109/WSC57314.2022.10015250.

Kashani, S. M., E. Yavari, and T. Khatibi. 2024. “Optimizing Emergency Department Resource Allocation Using Discrete Event Simulation and Machine Learning Techniques”. *Journal of Archives in Military Medicine* 11(4):1–8.

Kubat, M. 2017. *An Introduction to Machine Learning*. Cham: Springer Verlag.

Law, A. M., and W. D. Kelton. 2000. *Simulation Modeling & Analysis*. 3rd ed. New York: McGraw-Hill.

Mathew, V., T. Toby, V. Singh, B. M. Rao, and M. G. Kumar. 2017. “Prediction of Remaining Useful Lifetime of Turbofan Engine Using Machine Learning”. In *IEEE International Conference on Circuits and Systems*, 306–311.

MCA 2025. “Search and Rescue Helicopter Annual Statistics: Year Ending March 2025”. https://www.gov.uk/government/statistics/search-and-rescue-helicopter-statistics-year-ending-march-2025/search-and-rescue-helicopter-annual-statistics-year-ending-march-2025, accessed $10^{th}$ June 2025.

Mladenić, D., I. Bratko, R. J. Paul, and M. Grobelnik. 1994. “Using Machine Learning Techniques to Interpret Results from Discrete Event Simulation”. In *Proceedings of the European Conference on Machine Learning*, 399–402.

Olave-Rojas, D., and S. Nickel. 2021. “Modeling a Pre-Hospital Emergency Medical Service Using Hybrid Simulation and a Machine Learning Approach”. *Simulation Modelling Practice and Theory* 109:102302.

Ortiz-Barrios, M., A. Ishizaka, M. Barbati, S. Arias-Fonseca, J. Khan, M. Gul, *et al*. 2024. “Integrating Discrete-Event Simulation and Artificial Intelligence for Shortening Bed Waiting Times in Hospitalization Departments during Respiratory Disease Seasons”. *Computers & Industrial Engineering* 194:110405.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, *et al*. 2011. “Scikit-Learn: Machine Learning in Python”. *Journal of Machine Learning Research* 12:2825–2830.

Petchrompo, S., and A. K. Parlikad. 2019. “A Review of Asset Management Literature on Multi-Asset Systems”. *Reliability Engineering & System Safety* 181:181–201.

Pintelon, L., S. K. Pinjala, and A. Vereecke. 2006. “Evaluating the Effectiveness of Maintenance Strategies”. *Journal of Quality in Maintenance Engineering* 12(1):7–20.

Robinson, S. 2014. *Simulation: the Practice of Model Development and Use*. London: Palgrave MacMillan.

Shapley, L. S. 1953. “A Value for N-Person Games”. *Contributions to the Theory of Games* 2:307–317.

Si, X. S., W. Wang, C. H. Hu, and D. H. Zhou. 2011. “Remaining Useful Life Estimation – a Review on the Statistical Data Driven Approaches”. *European Journal of Operational Research* 213(1):1–14.

Swanson, L. 2001. “Linking Maintenance Strategies to Performance”. *International Journal of Production Economics* 70:237–244.

Theeuwes, N., G. J. van Houtum, and Y. Zhang. 2021. “Improving Ambulance Dispatching with Machine Learning and Simulation”. *Machine Learning and Knowledge Discovery in Databases* 4(21):302–318.

Štrumbelj, E., and I. Kononenko. 2014. “Explaining Prediction Models and Individual Predictions with Feature Contributions”. *Knowledge and Information Systems* 41:647–665.

Werner, C. 2023. “Simulation Input Modelling in the Absence of Historical Data for Decision Support during Crises: Experience with Assessing Demand Uncertainties for Simulating Walk-Through Testing in the Early Waves of COVID-19”. *Journal of the Operational Research Society* 74(2):489–508.

Zhang, J., P. Wang, R. Yan, and R. X. Gao. 2018. “Long Short-Term Memory for Machine Remaining Life Prediction”. *Journal of Manufacturing Systems* 48:78–86.

## AUTHOR BIOGRAPHIES

**CHRISTOPH WERNER** is a senior simulation consultant at Minitab. He joined in 2018 (then Simul8) and has delivered simulation projects in many different areas, ranging from the automotive, defense, food/beverage and energy industries to healthcare and other service sectors. He has an MSc and PhD in Operational Research/Risk Analysis from the University of Strathclyde, UK, and his research focuses, in particular, on modeling challenges in uncertainty quantification. His email is cwerner@minitab.co.uk.