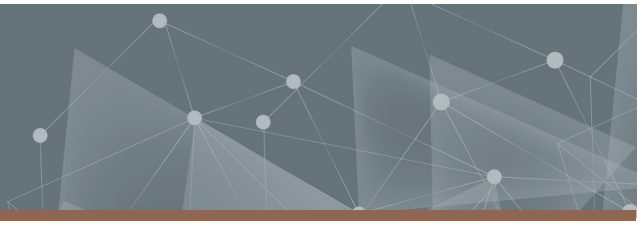
CHALMERS
UNIVERSITY OF TECHNOLOGY



Neural Networks for Predicting Fluid Filter Remaining Useful Life

Master's thesis in Systems, control and mechatronics

Lennard Tiltmann

DEPARTMENT OF INDUSTRIAL AND MATERIALS SCIENCE

CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's thesis 2025

Neural Networks for Predicting Fluid Filter Remaining Useful Life

Lennard Tiltmann

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Industrial and Materials Science
Chalmers University of Technology
Gothenburg, Sweden 2025

Neural Networks for Predicting Fluid Filter Remaining Useful Life
Lennard Tiltmann

Supervisors: Silvan Marti, Anirudha Das
Examiner: Björn Johansson

Master's Thesis 2025
Department of Industrial and Materials Science
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden
Telephone +46 31 772 1000

Typeset in LaTeX
Gothenburg, Sweden 2025

Neural Networks for Predicting Fluid Filter Remaining Useful Life
Lennard Tiltmann
Department of Industrial and Materials Science
Chalmers University of Technology

Abstract

This research addresses the challenge of estimating the Remaining Useful Life (RUL) of oil filters in industrial hydraulic systems using data-driven predictive maintenance. Focusing on a proprietary dataset characterized by a severely limited number of operational cycles and sparse laboratory measurements, the study evaluates traditional machine learning and deep neural networks under various feature engineering approaches. Findings reveal that for this constrained dataset, predictive accuracy is critically dependent on a single, dominant feature representing the filter's total workload. Consequently, RUL defined by processed oil volume proved to be a more robust and predictable target than one based on operational time. While complex feature engineering and models struggled with the limited data, the same methodologies demonstrated strong performance on comprehensive benchmark datasets. To overcome data limitations in the target application, a market study for inline particle sensors was conducted, identifying feasible technologies that could provide the high-frequency oil cleanliness data necessary for robust future RUL predictions. The study underscores the fundamental importance of sufficient, relevant data for successful predictive maintenance implementation.

Keywords: predictive maintenance, remaining useful life (rul), oil filters, sensor technology, machine learning, deep learning, time series data, data scarcity

List of Acronyms

1D-CNN	One-Dimensional Convolutional Neural Network
C-MAPSS	Commercial Modular Aero-Propulsion System Simulation
CMOS	Complementary Metal-Oxide-Semiconductor
EDA	Exploratory Data Analysis
GRU	Gated Recurrent Units
HI	Health Index
ISO	International Organization for Standardization
KPI	Key Performance Indicators
LDA	Linear Discriminant Analysis
LMCNN	Learnable Multiscale Convolutional Neural Network
LOOCV	Leave-One-Out Cross-Validation
LSTM	Long Short-Term Memory
MLP	Multi-Layer Perceptron
MPC	Membrane Patch Colorimetry
OEM	Original Equipment Manufacturer
PCA	Principal Component Analysis
PdM	Predictive Maintenance
PHM	Prognostics and Health Management
RH	Relative Humidity
RMSE	Root Mean Squared Error
RNN	Recurrent Neural Network
RUL	Remaining Useful Life
SHAP	SHapley Additive exPlanations
SSPD	Simple State-Based Prognostic Model for Filter Clogging
tsfresh	Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests

Nomenclature

General Mathematical and Statistical Symbols

n	Number of observations or samples
R^2	Coefficient of Determination score
$\mathbf{x}$	Input data vector
y_i	Actual (true) value for the i-th sample
$\hat{y}_i$	Predicted value for the i-th sample
$\bar{y}$	Mean of the actual (true) values

Machine Learning and Deep Learning Specific Symbols

σ	Non-linear activation function (e.g., sigmoid, ReLU)
$\odot$	Element-wise multiplication (Hadamard product)
tanh	Hyperbolic tangent function
$a_j^{(l)}$	Activation (output) of neuron j in layer l
b	Bias vector in a neural network layer
h_t	Hidden state of a recurrent unit at time step t
$\tilde{h}_t$	Candidate hidden state of a GRU unit at time step t
k	Kernel (filter) in a convolutional neural network
K	Size (length) of a 1D convolutional kernel
r_t	Reset gate of a GRU unit at time step t
U	Weight matrix associated with the hidden state in a recurrent unit
$\mathbf{w}$	Weight vector (e.g., an eigenvector in PCA)
W	Weight matrix associated with the input in a neural network layer
x_t	Input to a model at time step t
z_t	Update gate of GRU unit at time step t
z	Principal component (in PCA)

Physical Quantities and Domain-Specific Symbols

ΔP	Pressure drop
P	Measured pressure
Q	Measured flow rate
R_h	Hydraulic resistance

Contents

List of Figures

List of Tables

1

Introduction

1.1 Background

Fluids, such as lubricants and hydraulic oils, are essential for the proper functioning and longevity of a wide range of industrial machinery. They reduce friction and wear between moving components, transmit power, dissipate heat, and protect against corrosion [1]. Comprehensive management of these fluids, which is often termed Fluid Management, is therefore crucial for maintaining equipment reliability, operational efficiency, and overall system performance [2, 3]. Oil filters play an indispensable role in maintaining cleanliness by removing particulate contaminants and degradation byproducts. Implementing these practices can keep the oil in good condition add significant economic benefits [2].

Traditional maintenance strategies relied on reactive (run-to-failure) or preventive (time- or usage-based) schedules. While these maintenance approaches minimize upfront costs, they carry a high risk of catastrophic failure and unexpected downtime. In response to these limitations, predictive maintenance (PdM) has emerged as a more sophisticated, economically advantageous approach. This approach uses continuous monitoring of equipment conditions to anticipate potential failures and schedule maintenance proactively [4].

A key objective of PdM is estimating a system's remaining useful life (RUL), which provides a quantitative measure of when a component or system requires maintenance or replacement. This approach allows the precise scheduling of maintenance of filters, when needed.

Although oil properties can be precisely measured in a laboratory, this offline approach is time-consuming and expensive, considering the additional steps required for the analysis [5]. To overcome these limitations, online sensor technology has become essential for the successful implementation of PdM. These sensors allow continuous or high-frequency monitoring of critical fluid parameters directly in the operational system.

These parameters commonly include physical properties such as viscosity and temperature, chemical characteristics like acidity or total base number, the presence and concentration of contaminants such as water or wear particles, and other system conditions [5, 6].

The generation and accumulation of wear particles are primary contributors to equipment degradation and failure. Monitoring the concentration and character-

istics of these particles can provide direct insight into the wear rate and progression within lubrication systems [6, 7].

Time-series data acquired from various sensors can reveal important degradation trends that indicate a filter's decreasing performance and approaching end of life when analyzed effectively [8]. This data is essential for developing sophisticated prognostic models. Machine learning and deep neural network architectures, in particular, have shown significant promise in processing these complex datasets to identify degradation patterns and accurately predict the remaining useful life of various components [9].

1.2 Related Work

The field of predictive maintenance (PdM) and Remaining Useful Life (RUL) estimation for industrial components, particularly those involving fluid systems and filtration, has seen diverse approaches. This section reviews several recent studies that highlight relevant methodologies, from advanced sensor development and signal processing to application of machine learning models.

Tao et al. [10] present a wear condition warning method for wind turbine gearboxes centered around online oil monitoring in their work. Their system uses a high-performance machine vision sensor that can detect and categorize magnetic and nonmagnetic metallic wear particles down to 4 µm in size. Image features are extracted using two image segmentation algorithms. A key contribution is the LM-CNN architecture, a novel learnable multiscale convolutional neural network that integrates real-time online monitoring data with offline laboratory analysis results, including historical data. This approach demonstrated high fault detection accuracy, surpassing conventional machine learning methods such as k-nearest neighbors and random forests in their context.

Focusing on the degradation of filters, Skaf, Eker, and Jennions [11] proposed a Simple State-Based Prognostic Model for Filter Clogging (SSPD). Their work addressed the challenge of observing natural filter clogging by creating an artificial progression of the failure mode. The degradation level was divided into discrete states using k-means clustering. The RUL was then estimated based on the expected duration in the current and following health states. The study showed that this state-based prognostic method could accurately predict the RUL of the clogging filter under the defined and controlled laboratory conditions.

Lee et al. [12] addressed data-driven health condition assessment and RUL prognosis for liquid filtration systems. They introduced a Health Index (HI) for the filter, derived from sensor data, to quantify its condition. K-means clustering was then used to identify the performance degradation point from the HI trend. Various Recurrent Neural Network (RNN) architectures, including vanilla, stacked, and bidirectional Long Short-Term Memory (LSTM) networks, were employed to predict the future evolution of the HI. In their results, the bidirectional LSTM model provided the most accurate performance in predicting the HI and therefore the RUL of the filtration system.

Addressing the challenge of condition monitoring for complex hydraulic systems, Helwig, Pignanelli, and Schütze [13] developed a systematic approach for automated training of monitoring systems using multivariate statistics. In their approach, the authors analyzed various fault scenarios on a hydraulic test rig. Features were extracted from the raw sensor data, and their correlation with known fault characteristics was used to identify the most significant fault-specific features. Linear Discriminant Analysis (LDA) was then applied to classify fault conditions and their intensity.

1.3 Research Questions

This research builds upon the foundational understanding of predictive maintenance for fluid filtration systems, addressing specific challenges presented by real-world operational data and its multi-source nature. To investigate these complexities and opportunities, the following research questions have been defined to guide this study:

1. How do traditional machine learning models compare to deep neural networks in predicting oil degradation in fluid filtration dynamics based on the available data?

2. Which sensor parameters contribute the most to accurate predictions, and how can their importance be effectively identified?

3. What sensor technologies can be integrated into the existing measurement setup to improve the accuracy and reliability of filter change predictions?

1.4 Thesis Structure

This thesis is organized into five chapters. The Introduction chapter frames the context by providing an overview of Fluid Management and predictive maintenance, reviewing relevant literature, and outlining the addressed research questions. The Theory chapter explains fundamental concepts and principles relevant for this study, including the definition of Remaining Useful Life, sensor-based monitoring techniques, and the theoretical foundations of the predictive modeling approaches employed. The Methods chapter introduces the datasets that were used, elaborates on the data preprocessing and feature engineering steps, and describes the selected machine learning and deep learning models as well as the implemented evaluation strategies. The Results chapter presents the findings from the exploratory data analysis, the performance of the predictive models across various scenarios and datasets, and the insights gained from the particle sensor study. Finally, the Conclusion chapter summarizes key contributions of the research, discusses the implications of those findings, acknowledges the limitations encountered, and proposes directions for future work.

1.5 Ethics

This thesis made limited use of artificial intelligence (AI) tools to support the research process. These tools were not used to derive the scientific conclusions or results. Applications included writing assistance, such as language refinement, proofreading, rephrasing paragraphs, summarizing text, or code debugging. The specific tools used were Google's large language model Gemini, Deepl Write and Github Copilot inside Visual Studio Code. All generated content by AI tools was critically reviewed and verified.

2

Theory

2.1 Sensor-Based Monitoring in Predictive Maintenance

Sensor-based monitoring is a key component of modern predictive maintenance, providing real-time data needed to assess the system condition.

2.1.1 Online vs. Offline Monitoring

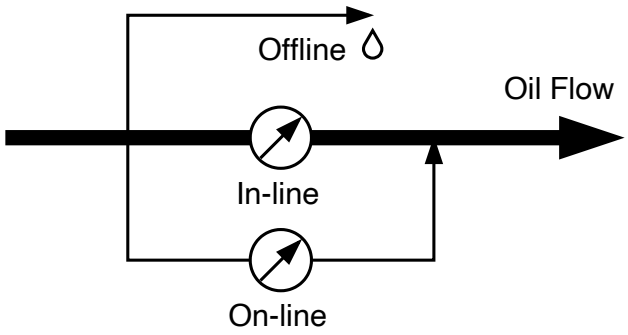


Figure 2.1: Oil particle analysis monitoring location, adapted from [14].

Offline monitoring involves collecting data periodically and analyzing it outside of the system, often in a laboratory setting. This approach allows for detailed and precise analysis using specialized equipment. However, it is typically time-consuming, costly due to manual effort and transport, and provides only infrequent samples of the system's condition.

Online monitoring allows continuous or high-frequency data acquisition directly from the system during the operation. The sensors are directly exposed to the flow of the fluid, either in the main pipe (in-line) or in a bypass (on-line). Common sensor types for monitoring fluid parameters include sensors for temperature, pressure, flow rate, and relative humidity. The concept is presented in Figure 2.1. In this work, the term *online* covers both *on-line* and *in-line* concepts; however, in existing literature, these terms are sometimes used interchangeably.

2.1.2 Principles of Particle Contamination Sensing

ISO 4406

ISO 4406 is an international standard for expressing the level of particulate contamination in fluids [15]. It defines a code based on the particles greater than specified sizes (4 µm, 6 µm, and 14 µm) in a fixed volume of 1 mL. The code consists of three numbers representing the contamination level for these three size ranges. Although these numbers do not directly represent particle counts, they indicate concentration ranges, allowing for easy handling and classification.

Light Extinction

Light extinction is a principle used in optical particle counters. It works by passing a focused beam of light through a flow cell containing the fluid sample. As particles pass through the beam, they obstruct or scatter the light, causing a reduction in the light intensity reaching a detector. The magnitude of the change in light intensity is proportional to the particle size. Counting these events allows for determining particle size distribution and concentration [16].

Digital Imaging

Digital imaging sensors capture images of particles on a detector array (typically CMOS detector). A light source casts the light through the oil sample or flow, projecting an image on the sensor area. Two main concepts exist for the optical setup. Direct imaging systems utilize additional optics to project an image onto the sensor, enabling detailed information about particle size distribution, shape, and soot content [17]. However, these systems can be limited by a narrow field of view and depth of field. Lensless imaging techniques, on the other hand, involve simpler optics, often using an aperture (pinhole) near the light source. This approach allows for higher flow rates compared to some direct imaging setups while still providing particle size, shape, and count information [18].

2.1.3 Remaining Useful Life (RUL)

Remaining Useful Life (RUL) is a concept in prognostics and predictive maintenance, representing the estimated time or cycles left before a component or system requires maintenance, repair, or replacement [19]. RUL can be expressed in various units, such as hours of operation, number of cycles, or accumulated volume throughput, depending on the system and application. Although RUL is often simplified to a linearly decaying value for modeling purposes, more precise representation can involve complex, non-linear degradation paths influenced by multiple real-world factors [20].

2.1.4 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is a technique to reduce dimensionality of high-dimensional feature spaces. It transforms a set of possibly correlated variables into a smaller set of uncorrelated variables called principal components [21]. The

first principal component captures the largest possible variance in the data, and subsequent components capture the remaining variance in orthogonal directions. The number of components can be selected as a tradeoff between explained variance and the number of components in the feature space.

The first principal component z_1 for a data vector $\mathbf{x}$ (mean-centered) is given by:

$$z_1 = \mathbf{w}_1^T \mathbf{x}$$

where $\mathbf{w}_1$ is the eigenvector corresponding to the largest eigenvalue of the covariance matrix of $\mathbf{x}$.

2.2 Predictive Modeling Techniques

RUL estimation from sensor data can be achieved using a wide range of predictive modeling techniques, spanning from traditional machine learning methods to advanced deep learning architectures.

2.2.1 Traditional Machine Learning Models

Ensemble Tree-based Methods

Ensemble tree-based methods combine the predictions from multiple decision trees to improve overall predictive performance and robustness [22]. The core idea of this apporach is to aggregate the decisions of many individual models for a more accurate and stable prediction than individual model could achieve. These methods can be broadly categorized into bagging and boosting techniques, which differ in how the individual trees are built and combined.

The ensemble prediction $\hat{y}_{\text{ensemble}}$ is often a weighted average of individual model predictions $\hat{y}_k$:

$$\hat{y}_{\text{ensemble}} = f(\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_N)$$

where N is the number of models in the ensemble.

Random Forests

Random Forests is an ensemble method based on bagging, specifically applied to decision trees. It constructs multiple decision trees during training, each trained on a random subset of the training data and considering only a random subset of features at each split point. For regression tasks, the final prediction is typically the average of the predictions across the individual trees [23].

Boosting Algorithms

Boosting is another ensemble technique that builds models sequentially, where each new model attempts to correct the errors made by the previous ones. It iteratively focuses on misclassified or poorly predicted samples and giving them higher weight in following training steps. Popular boosting algorithms include AdaBoost (Adaptive Boosting) and Gradient Boosting [24, 25].

Multi-Layer Perceptron (MLP)

A Multi-Layer Perceptron (MLP) is a fundamental type of feedforward artificial neural network. The MLP consists of an input layer, one or more hidden layers, and an output layer. Each layer is composed of artificial neurons that are connected to neurons in the following layer with weighted connections. The data flows only in one direction, from input to output. Non-linear activation functions are applied to the output of each neuron to enable the network to learn complex patterns. The network learns by adjusting the weights and biases through an optimization algorithm, typically using backpropagation [26].

The output $a_j^{(l)}$ of neuron j in layer l with activation function σ is given by:

$$a_j^{(l)} = \sigma\left(\sum_i w_{ji}^{(l)} a_i^{(l-1)} + b_j^{(l)}\right)$$

where $w_{ji}^{(l)}$ are weights, $a_i^{(l-1)}$ are inputs from layer $l-1$, and $b_j^{(l)}$ is the bias.

MLP's are foundational neural network structures. In this work, due to the relatively small number of neurons and layers used, the MLP is categorized as a traditional machine learning method rather then deep learning model.

2.2.2 Deep Learning Models for Sequential Data

1D Convolutional Neural Networks (1D-CNN)

1D Convolutional Neural Networks (1D-CNNs) are well-suited for processing sequential data like time series [27]. They use convolutional layers with 1D filters (kernels) that slide across the input sequence, allowing them to detect local patterns. These convolutional layers are typically followed by pooling layers (e.g., max pooling or average pooling) which reduce the spatial dimension and provide a more robust representation. 1D-CNNs can effectively capture dependencies within local windows of the time series.

A 1D convolution operation $y[t]$ on an input sequence $x[t]$ with a kernel $k[i]$ is given by:

$$y[t] = \sum_{i=0}^{K-1} x[t-i]k[i]$$

where K is the kernel size [28].

Gated Recurrent Units (GRU)

Gated Recurrent Units (GRUs) are a type of recurrent neural network (RNN) designed to address the vanishing gradient problem [29] and capture long-term dependencies in sequences. GRUs use gating mechanisms, specifically an update gate and a reset gate, to control the flow of information and regulate the amount of the past memory to be updated or forgotten. This allows GRUs to maintain relevant information over long sequences.

Relevant equations for a GRU at time step t involve computing the update gate z_t and reset gate r_t:

$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$$
$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$$

A hidden state $\tilde{h}_t$ is computed using the reset gate:

$$\tilde{h}_t = \tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h)$$

Finally, the new hidden state h_t is a linear combination of the previous state and the candidate state, controlled by the update gate:

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot \tilde{h}_t$$

where x_t is the input, h_{t-1} is the previous hidden state, W and U are weight matrices, b are bias vectors, σ is the sigmoid function, tanh is the hyperbolic tangent, and $\odot$ denotes element-wise multiplication [30].

2.3 Model Evaluation Strategies

Robust evaluation strategies and appropriate performance metrics are critical for the accuracy and generalizability of predictive models.

Cross validation

Cross-validation is a technique used to assess how the results of a statistical analysis generalize to an independent dataset. The principle involves dividing the data into subsets; the model is trained on a part of and validated on the remaining unseen data. For small datasets, Leave-One-Out Cross-Validation (LOOCV) uses each data point once for validation. When data samples belong to groups that should not be split between training and validation sets (e.g., all samples from one cycle should be in the same split), GroupKFold cross-validation from the scikit-learn library [31] is a suitable strategy, ensuring samples from the same group are kept together and no information is leaked across the splits.

2.3.1 Performance Metrics for RUL Regression

R^2 (Coefficient of Determination)

The R^2 score measures the proportion of the variance in the dependent variable that is predictable from the independent variables [32]. It provides an indication of how well the model fits the data, ranging from 0 (no variance explained) to 1 (all variance explained). The score can be negative if the model predicts worse than the average value.

The R^2 score is calculated as:

$$R^2 = 1 - \frac{\text{Sum of Squares of Residuals}}{\text{Total Sum of Squares}} = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

where y_i are the actual values, $\hat{y}_i$ are the predicted values, and $\bar{y}$ is the mean of the actual values.

Root Mean Squared Error (RMSE)

Root Mean Squared Error (RMSE) is a frequently used metric to measure the differences between values predicted by a model and the values observed. It represents the square root of the average of the squared differences between predicted and actual values, giving a measure of the typical prediction error magnitude.

The RMSE is calculated as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

where y_i are the actual values, $\hat{y}_i$ are the predicted values, and n is the number of observations.

3

Methods

3.1 Data Description

This study utilizes a combination of proprietary and publicly available datasets to develop and validate the predictive models. The following sections describe the primary dataset central to this research, as well as two complementary benchmark datasets used for comparative analysis.

3.1.1 Primary Oil Filter Dataset

The core of this research is based on a proprietary dataset collected from the hydraulic systems of two different hydraulic presses. The data relates specifically to the monitoring of custom filtration systems within these machines operating under normal production conditions. The oils used in the machines are *Holst 22* and *Holst 46* produced by *Q8Oils*, which are mineral zinc-free hydraulic oils.

This dataset, referred to as the Primary Oil Filter Dataset in the following, is used for developing predictive maintenance models for oil filter replacement in this specific application.

The data consists of multi-source time-series information captured over several filter life cycles. Each cycle captures one sequence starting with the filter change and ending with the replacement of the filter. This time period ranges from 51 to 200 days in the dataset. The data streams originate from two distinct sources with different collection methodologies and frequencies:

Sensor Data

Recorded continuously over the filter life cycles, the sensor data provides regular measurements of the oil condition. The key parameters monitored via inline sensors are:

- Temperature (°C)
- Relative Humidity (%)
- Flow Rate (L/min)
- Pressure (bar)

These sensor readings are recorded at a relatively high frequency (typically one data point per hour), providing a detailed time-series profile of each filter cycle.

Laboratory Measurements

In addition to the sensor data, laboratory analyses are available for discrete times in the cycle. For the analysis, samples were manually taken from the oil tanks and therefore considered offline measurements. These measurements can provide complementary information about the oil degradation state and contamination level, but are collected at significantly less frequent and irregular intervals compared to the sensor data. For each filter cycle, only a few samples were obtained. The number of measurements per cycle ranges from three to seven. The parameters available in the analysis data are:

- Membrane Patch Colorimetry (MPC)
- Particle Count (as ISO 4406 and as direct particle count values)
- Water Content (ppm)

The particle count information is given in three different particle size classes: for particles of 4 µm, 6 µm, and 14 µm.

Dataset Limitations

A significant characteristic and limitation of the Primary Oil Filter Dataset is the number of recorded operational cycles for filter degradation monitoring across the two hydraulic systems. The dataset contains data from a total of only four complete operating filter cycles, with one system recording three cycles and the other system one. This extremely low number of cycles, combined with potential heterogeneity between the two different machines, significantly limits the amount of training data available and introduces substantial challenges for developing robust and generalizable predictive models.

Adding to these challenges, while potentially providing crucial information on oil degradation and contamination, the laboratory measurements are collected at very low and irregular frequencies, with only three to seven samples available per filter cycle. This sparsity severely restricts the ability to capture detailed degradation trajectories from these parameters and limits their effective integration into time-series models. The utility of the dataset is further impacted by other real-world complexities, including discontinuities caused by machine downtime, maintenance activities, and production schedule variations (such as holidays), as well as potential noise and inconsistency inherent in sensor and laboratory measurements.

3.1.2 Complementary Datasets

The inclusion of these additional data sources served several purposes. First, all datasets were used to test the same models and approaches, allowing for comparisons across datasets. Second, statements can be made about the robustness and generalizability of the developed methods in these scenarios. Finally, even in the face of limited success with the primary dataset due to its inherent limitations, these complementary datasets provided a viable way to validate the overall conceptual approach to predictive maintenance and demonstrate the potential of the chosen methods when applied to more comprehensive data. Two main complementary datasets

were selected for these purposes: the C-MAPSS aircraft engine simulator data and the Preventive to Predictive Maintenance dust filter dataset.

C-MAPSS Aircraft Engine Simulator Data

The Commercial Modular Aero-Propulsion System Simulation (C-MAPSS) dataset, developed by NASA, is a widely recognized benchmark for prognostics and health management (PHM) research, particularly for Remaining Useful Life (RUL) estimation [33]. Over 70 publications were released analyzing this dataset [34]. It consists of simulated run-to-failure data for a fleet of turbofan engines. These engines are fueled by gas and generate power through the combustion process, which drives a fan that pushes air backward, generating forward thrust. Each data instance is a multivariate time series representing an engine's degradation trajectory, including 21 sensor readings and 3 operational settings.

This dataset was chosen due to its extensive use in academic literature, providing a rich source of established methodologies and RUL prediction models that could be adapted or serve as a benchmark. Although the application domain differs significantly from the oil filter scenario, the C-MAPSS Dataset presents a similar problem structure: predicting RUL based on time-series sensor data from a degrading system.

The C-MAPSS Dataset is divided into four subsets (FD001, FD002, FD003, FD004), each characterized by different operating conditions and fault modes. For this project, only the FD001 subset was utilized, specifically its training data, which contains 100 engine degradation trajectory cycles.

Not the complete feature set was selected for the predictions in this project. In the FD001 subset, some features are completely stationary, providing no additional information for the model. Instead, only a part of the features were selected. This is a common practice, and the same feature set was selected in multiple publications [35]. The resulting dataset consists of 14 sensors from the original 24 features (21 sensors and 3 operational settings).

Selected sensor numbers: 2, 3, 4, 7, 8, 9, 11, 12, 13, 14, 15, 17, 20, and 21.

Preventive to Predictive Maintenance Dataset

To explore a system more analogous to the oil filtration problem, the "Preventive to Predictive Maintenance" dataset was selected [36]. Due to its long and unspecific name, the dataset is referred to as the "Dust Filter Dataset" in the following text. This dataset originates from a test bench designed to study the degradation (clogging) of filters during the separation of solid dust particles from a gas stream. This scenario shares more similarities on the physical level with the original oil filter application, compared to the C-MAPSS engine data.

The dataset was published by Hagmeyer, Mauthe, and Zeiler [36] as part of an effort to provide publicly available data for industrially relevant PHM scenarios. The training section of this dataset contains 50 run-to-failure trajectories from the air filtration test bench. Key recorded parameters include differential pressure across the filter, flow rate, and particle feeder velocity, along with characteristics of the test dust and filter media.

The included features for this project were:

- Differential Pressure (Pa)
- Flow rate
- Dust feed rate
- Dust type

The dust type was one of three different dust types, defined by the ISO 12103-1 standard: A2 Fine Test Dust, A3 Medium Test Dust, and A4 Coarse Test Dust. Since this information was available in one column as a string, the dust type was one-hot encoded to three binary features. Furthermore, the time information is given as a decimal number increasing in 0.1-time steps. For consistency, the time column was multiplied by a factor of ten, resulting in integer numbers, aligning with the other two datasets.

This dataset, with its larger number of 50 cycles for the test set and focus on filter degradation, provided a valuable resource for further testing and evaluation of the developed predictive maintenance framework.

3.2 Data Preprocessing

Data preprocessing was a crucial initial step to transform the raw, multi-source time-series data into a suitable format for analysis and predictive model development.

3.2.1 Data Handling

The raw data for the Primary Oil Filter Dataset was initially stored in separate tabular formats for sensor readings and laboratory measurements. Data was loaded and managed using Python's pandas library, transforming it into DataFrame structures for further manipulation. Unnecessary metadata was excluded during this process.

The sensor data tables contained continuous time-series recordings from the inline sensors (Temperature, Relative Humidity, Flow Rate, Pressure) along with associated timestamps and system identifiers. A cumulative oil volume value was also available, capturing the total throughput over time.

Laboratory measurements (MPC, Particle Count, Water Content) were recorded in a separate table. Each entry was associated with a specific sampling date. This table also included metadata like the filter change date, which was necessary for defining filter life cycles. Additional information about the system's running mode at the time of sampling (primarily recorded as fixed flow or pressure values for this dataset).

3.2.2 Data Cleaning and Preparation

Due to external factors, the dataset might include data points that are not representative of the filter degradation process. It was observed that, at the beginning of some cycles, the sensor values were zero. This might indicate periods where the

system was not yet operational, and therefore these initial data points do not reflect the actual oil parameters. Furthermore, it was found that for some data points, the cumulative flow was lower than the previous values. The origin of this behavior probably lies within the data collection method: The data is sent via a wireless connection and logged on an external device, with the timestamp created when the data is received. Therefore, the initial constant sensor values, as well as the delayed data points, were excluded from the dataset.

Further inconsistencies in the data logging needed to be resolved. For the temperature recordings, some logged values were scaled down by a factor of 10. These data points were identified as low temperature and scaled up to the correct value.

3.2.3 Alignment and Resampling

A key challenge resulting from the multi-source nature of the dataset was the differing sampling rates and misalignment between the continuously recorded sensor data and the irregularly sampled laboratory measurements. To create suitable unified time series for analysis, the data needed to be integrated and aligned.

To overcome this issue, the laboratory data was interpolated. Simple linear interpolation was chosen to estimate the values for the laboratory measurements in between the actual samples. This technique provided a continuous representation of MPC, particle count, and water content parameters across the filter cycle, allowing for these parameters to be assigned to the corresponding sensor values. Other interpolation methods, like forward or backward filling, were considered. These particular methods would result in a stepwise-constant trajectory of the values. However, at a particular time step, no information about the slope would be available. Cubic interpolation would add additional complexity, and boundary conditions would need to be handled. Nevertheless, any kind of interpolation introduces some degree of uncertainty, as the true behavior of the parameters between measurement points remains unknown and is only approximated by the chosen mathematical model.

Both data sources needed to be combined into a single DataFrame. Since the cumulative oil volume was available in both data tables, the merging was performed based on this parameter, creating a single DataFrame containing both sensor data and laboratory measurements.

Following the alignment step, all time-series data streams were resampled to a consistent one-hour interval. This interval was selected because the majority of the data points were recorded within a time period of approximately one hour.

Resampling all data to a consistent, uniform time frame served to combine all variables into a single, synchronized, time-indexed DataFrame. This uniformity is beneficial as it allows for consistent analysis across all sensor and laboratory variables and simplifies subsequent feature engineering and model development processes. The cumulative oil volume was included as a feature within this combined DataFrame.

3.3 Feature Engineering

To enhance the predictions of the models, the raw time-series data were enriched through various feature engineering techniques. These include the creation of domain-specific features based on physical principles, the inclusion of metadata, and automated feature extraction methods.

3.3.1 Domain-Specific Features

In addition to the raw sensor and laboratory measurements, several domain-specific features were engineered to better capture the underlying physical processes of oil degradation and filter clogging. These features are derived from existing data based on an understanding of hydraulic system behavior and filter performance principles, aiming to provide more direct indicators of the filter's health status. Domain-specific features were derived only for the Primary Oil Filter Dataset.

Cumulative Oil Volume

The cumulative oil volume, representing the total volume of oil that has passed through the filter up to a given point in time, was included as a key feature. This feature directly quantifies the total throughput or operational workload experienced by the filter. The value is provided in the dataset already and does not need to be calculated separately.

Particle Count Ratios

To capture shifts in the particle size distribution, which can be indicative of filter performance changes or specific wear mechanisms, ratios between different particle size counts were calculated. The following ratios were derived from the laboratory particle count data:

- Ratio of 4 µm to 14 µm particles
- Ratio of 4 µm to 6 µm particles
- Ratio of 6 µm to 14 µm particles

Changes in these ratios over time can provide insights beyond simple increases in total particle counts. For instance, a disproportionate increase in smaller particles relative to larger ones might suggest a decline in the filter's efficiency in capturing finer contaminants as it approaches saturation.

Hydraulic Resistance Proxy

The ratio of pressure to flow rate was engineered as a proxy for the hydraulic resistance of the filter. Hydraulic resistance (R_h) is a measure of the opposition to flow through a component. For a filter, it is expected to increase as the filter element becomes clogged with contaminants. Conceptually, this can be related by analogy to Ohm's Law, where the pressure drop (ΔP) across the filter is proportional to the flow rate (Q) and the hydraulic resistance:

$$\Delta P = Q \cdot R_h$$

Thus, the resistance can be estimated as:

$$R_h \approx \frac{P}{Q}$$

where P is the measured pressure required to drive flow through the filter and Q is the flow rate. An increasing trend in this P/Q ratio could be a strong indicator of filter clogging and impending failure, as more pressure is needed to maintain flow.

Cumulative Contaminant and Degradation Indicators

To represent the total accumulation of contaminants and degradation byproducts over the filter's operational life, cumulative sums were calculated for several laboratory measurements. These include:

- Cumulative Membrane Patch Colorimetry (MPC) value
- Cumulative particle count for 4 µm particles
- Cumulative particle count for 6 µm particles
- Cumulative particle count for 14 µm particles

These cumulative features integrate the exposure to contaminants or degradation products over time. Unlike the direct measurements, which can fluctuate, the cumulative sum provides a monotonically increasing measure of the total containment-related parameters the filter has encountered.

Running Mode Derived Features

The hydraulic system operates under different *running modes*, where it attempts to maintain either a specific target pressure or a specific target flow rate. The active mode (pressure control or flow control) and its corresponding target value are known for each operational phase. To leverage this information, two new features were engineered:

- **Pressure Operation Feature:** If the system is in pressure control mode, this feature is calculated as the difference between the target and the actual pressure. It represents the deviation from the pressure setpoint.

  If the system is in flow control mode, this feature simply takes the value of actual pressure.

- **Flow Operation Feature:** If the system is in flow control mode, this feature is calculated as the difference between the target flow and the actual flow. It represents the deviation from the flow setpoint.

  If the system is in pressure control mode, this feature simply takes the value of actual flow value.

These two features thus provide insights into both the system's success in meeting its active control target and the resulting value of the other primary parameter (pressure or flow) which is not being directly targeted in that specific mode.

3.3.2 Meta Data Related Features

In addition to the above engineered features, which were derived directly from the sensor or laboratory data, additional metadata was included in the dataset. As previously mentioned, the filter cycles exhibited real-world complexities, some of which are hard to include as additional information for the models. Information about machine downtime and maintenance was not consistently available for the filter cycles, and therefore could not be included.

To include information about other time-based events, the logged time of the sensor value was used. The timestamp was not directly included; instead, it was used to create cylindrical features, which is a common technique in time-series prediction [37, 38]. The period was set to one week, relating to the weekday and the time of the day. Another cycle spanned the full year, capturing seasonal information.

Additionally, information about the Swedish summer break and related factory downtime was included (as a binary feature, which is 1 or 0 accordingly). The actual times for the break could be observed directly from the data through rapid changes in the sensor values. For additional holidays, the Python holidays library was used. The library allows for retrieving public Swedish holidays for a given time. This information was used to create an additional binary feature.

Since the filter cycles originate from two different industrial machines and settings, this information is crucial for prediction. An additional binary feature was created for this information. Further properties of the systems (type of the machine, oil type, etc.) were not included, since they would be represented either through constant features across all filter cycles, or be identical to the previous one. If more filter cycles from different systems become available in the future, adding features for this information will need to be considered.

In total, ten metadata features were created in this step.

3.3.3 Automated Feature Extraction

The domain-specific features were manually derived as useful potential characteristics. Other relations and sensor characteristics that can be directly relevant might not be covered by manual selection alone. Therefore, an automatic feature generation method was chosen.

Automated feature extraction was used to methodically generate a wide range of characteristics from the preprocessed time-series data. This method uses algorithms to create many features from each time series. It can help find complex patterns related to filter degradation that might be missed or difficult to identify. For this purpose, the Python package tsfresh (Time Series FeatuRe Extraction on Basis of Scalable Hypothesis tests) was utilized [39].

This automated feature extraction strategy, which included time-based windowing, was specifically developed for the traditional machine learning models evaluated in this study. Deep learning models, discussed later, often have their own ways to learn features from sequence data and thus were typically fed with less processed time-series segments.

These feature extraction methods need to be applied to a time-series section. Applying it to the full cycle would add only one additional data point per cycle. For the four cycles of the Primary Oil Filter Dataset, this would add only four additional data points. The same value could then be added to all data points of each cycle, adding no extra information within each cycle. Instead, a window approach was applied, creating subsections (windows) of the cycle to which the feature generation function was applied. This process was applied consistently across the primary oil filter dataset as well as the complementary datasets to ensure comparability. Two different windowing methods were implemented:

- **Fixed-Size Sliding Window:** For every time step, a window of a fixed length was created. This window covered the data points immediately before and up to that time step. tsfresh was then applied to the sensor and interpolated laboratory data within this rolling window. This method aims to capture recent trends and patterns within a specific period.
- **Expanding Window:** For every time step (after an initial period corresponding to a minimum window size), this window included all data points from the start up to and including the current time step. tsfresh was then applied to this gradually growing piece of data. This method is designed to find overall effects and long-term trends that build up from the filter's start of use.

For both windowing methods, tsfresh was set up to calculate many different time-series features. These features are based on common methods from statistics, time-series analysis, and signal processing. The package provides different preconfigured sets of features. The MinimalFCParameters setting in tsfresh was selected as the main configuration. This setting gives a basic but useful group of features that are generally quick to calculate. This group includes, for instance, features for the median, mean, length, standard deviation, variance, root mean squared, maximum, absolute maximum, and minimum [40]. To further extend the feature set for monitoring filter degradation—where total effects and overall signal energy can be important—the MinimalFCParameters set was expanded by two metrics: sum of values and absolute signal energy. This ensured that features calculating the cumulative sum and the energy were included for each input time series in every window created.

Some methods from the MinimalFCParameters created identical features for all samples (e.g., the length for the fixed-sized window). These features contain no information about the underlying process. Therefore, an additional step was introduced to remove these features after creation.

tsfresh was applied to the primary dataset, as well as the complementary datasets with the same configuration. Due to the varying number of features in these datasets, the amount of the resulting extracted features differs considerably. For the Primary Oil Filter Dataset, the method was applied to the sensor and laboratory data including the domain-specific extended features. The metadata-related features were not included in the generation step, but added later to the dataset.

This larger set of features, while potentially containing a large amount of information, often has features that are repetitive or very similar to each other. This necessitated dimensionality reduction methods.

Table 3.1: Number of features in resulting datasets with tsfresh applied.

Dataset	Number of Features
Primary Oil Filter Dataset	290*
C-MAPSS Dataset	233
Dust Filter Dataset	61

* *Includes 10 metadata-related features*

3.3.4 Dimensionality Reduction

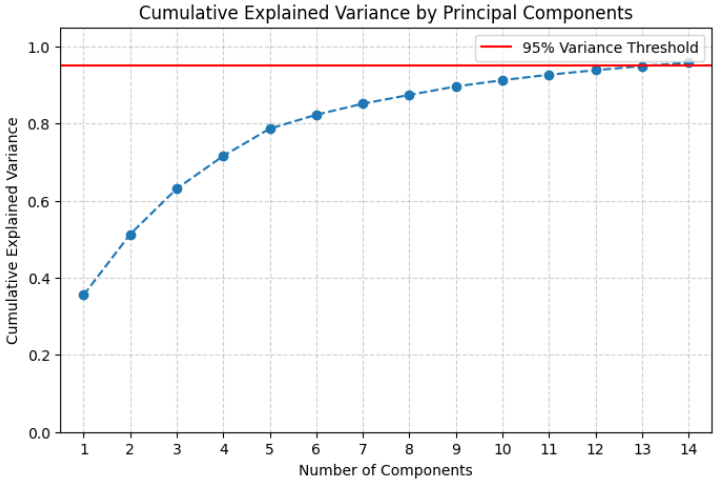
Following the automated feature extraction process with tsfresh, which significantly expanded the feature space for all datasets (as shown in Table 3.1), a dimensionality reduction step was applied to the data. The large number of generated features increases the risk of multicollinearity (where features are highly correlated with each other) and the curse of dimensionality, potentially leading to model overfitting and increased computational cost.

Principal Component Analysis (PCA) was selected for this purpose. PCA is a standard technique used to reduce the dimensionality of datasets containing many variables [41, 42]. It works by transforming the original set of potentially correlated features into a new, smaller set of linearly uncorrelated variables called principal components. These components are ordered such that the first few capture the majority of the variance present in the original data. The main goals of using PCA here were to reduce the number of input features for the predictive models and to handle the multicollinearity often introduced when generating numerous statistical features from time-series windows. Applying PCA to feature sets expanded through automated extraction is a documented approach, including for the C-MAPSS Dataset [43].

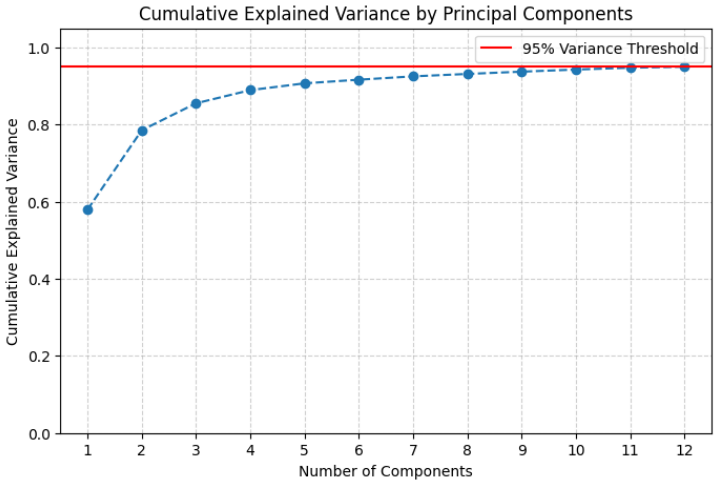
PCA was applied individually to the feature sets generated by tsfresh for the Primary Oil Filter Dataset, the C-MAPSS Dataset, and the Dust Filter Dataset. Before applying PCA, all features were standardized (scaled to have zero mean and unit variance). This ensures that features with larger values do not dominate the analysis. The number of principal components kept for each dataset was decided based on the cumulative explained variance. A threshold of 95% was chosen, meaning principal components were selected until they collectively explained at least 95% of the variance in the original tsfresh-generated feature set.

Figure 3.1 illustrates the relationship between the number of principal components selected and the cumulative variance explained for each dataset.

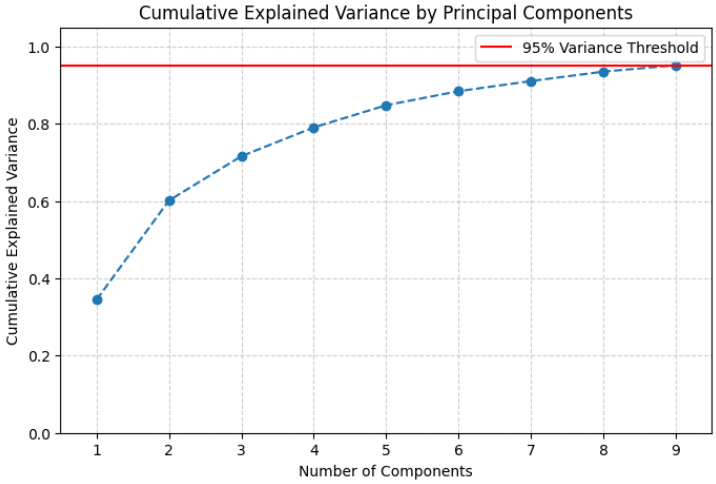
This application of PCA significantly reduced the dimensionality of the automatically extracted features for all datasets, producing a more compact and uncorrelated set of variables. These principal components then served as inputs for the predictive models. It should be noted that the 10 metadata-related features for the Primary Oil Filter Dataset were not included in the PCA transformation; they were appended to the resulting principal components afterwards.



(a) Primary Oil Filter Dataset



(b) C-MAPSS Dataset



(c) Dust Filter Dataset

Figure 3.1: Number of principal components with resulting cumulative explained variance.

3.4 Predictive Modeling

This section details the development of the predictive models, from defining the target variable to selecting and configuring the algorithms. The approach was designed to facilitate a systematic comparison of different modeling techniques across all datasets.

3.4.1 Defining the Target Variable

This project aims to predict the remaining useful life of the oil filter. Therefore, the remaining useful life *is* the target of our prediction. It is a value that can be calculated or is given for every data point in the dataset. The developed models are trained to predict this value, based on the training data and data samples. This defines the core task as a supervised regression problem.

Primary Oil Filter Dataset

For the Primary Oil Filter Dataset, the RUL was conceptualized and calculated in two distinct ways. Both definitions are considered meaningful within the operational context of industrial fluid filtration systems: time-based RUL is intuitive for maintenance scheduling, while volume-based RUL directly reflects the cumulative stress and particle load on the filter. This dataset, however, underlies significant inconsistencies and real-world complexities (such as machine downtime, maintenance activities, schedule variations, and the impact of events like holidays, as detailed in Section 3.1.1). Consequently, defining the true, complex, non-linear degradation trajectory and corresponding RUL precisely is extremely difficult based on the available data. The linear RUL definitions adopted here, both time-based and volume-based, therefore represent a pragmatic simplification of this complex process. The differences in predictive performance for these two linear target variable definitions will be investigated as part of the model evaluation.

The two RUL definitions are:

- **Time-based RUL (in hours):** This definition quantifies the remaining operational time until the filter is expected to require replacement. For each data point within a filter's operational cycle, the time-based RUL is calculated as the difference between the total duration of that specific cycle (from installation to replacement) and the elapsed operational time up to that data point. This results in a linearly decreasing RUL from the beginning of the cycle to zero at the end.
- **Volume-based RUL (in liters):** Assuming that filter degradation is heavily influenced by the volume of oil processed, a volume-based RUL was also defined. This represents the remaining volume of oil that the filter can process before replacement is necessary. Similar to the time-based RUL, it is calculated for each data point as the difference between the total cumulative oil volume recorded for the entire filter cycle and the cumulative oil volume up to that specific data point. This also follows a linearly decreasing RUL profile when viewed against cumulative volume.

For both definitions, an RUL value is computed for each hourly resampled data point within every filter cycle. These values are the target variable that machine learning and deep learning models are trained to predict.

Target Variable for Complementary Datasets

A common practice in the C-MAPSS literature is to define the RUL target variable using a piecewise linear function, often capping the maximum RUL at a certain value to reflect that significant degradation typically occurs closer to the end-of-life. However, for this project, a purely linear RUL definition (decreasing linearly from the maximum lifetime to zero) was adopted for the C-MAPSS data. This decision was made to maintain consistency with the RUL definition applied to the Primary Oil Filter Dataset. Furthermore, this approach, recognizing that a perfect, non-linear representation of RUL cannot always be assumed due to the data limitations and inconsistencies in that real-world case study, was adopted.

Similarly, for the Preventive to Predictive Maintenance (Dust Filter) Dataset, which includes run-to-failure trajectories for air filters on a test bench, the RUL for each filter degradation cycle was derived from its total operational lifespan until filter clogging. The RUL is defined as the remaining operational time, decreasing linearly from the start of the test run to zero when the filter is considered to have reached its end-of-life.

By employing a consistent linear RUL definition across all datasets, the approach aims to provide a clearer comparison of the feature engineering techniques and modeling approaches under similar target variable assumptions.

3.4.2 Selected Models

A key objective was to evaluate model architectures that could potentially be applied across all three datasets (Primary Oil Filter, C-MAPSS, Dust) to facilitate comparison. To ensure the chosen models are relevant for RUL prediction and allow for comparison with established benchmarks, the selection process considered models frequently applied and documented in academic literature, particularly those used with the extensively researched C-MAPSS dataset. The models were implemented using Python, leveraging the scikit-learn library [31] for traditional machine learning approaches and Keras [44] with a TensorFlow [45] backend for sequential deep learning models.

Four types of traditional machine learning models were selected for evaluation using the scikit-learn library:

- **Random Forest Regressor**
- **AdaBoost Regressor**
- **Gradient Boosting Regressor**
- **Multi-Layer Perceptron (MLP) Regressor**

The three tree-based ensemble models (Random Forest, AdaBoost, Gradient Boosting) were chosen partly because their implementations readily provide feature im-

portance scores, offering potential insights into the predictive power of the various input features. Initial tests involving hyperparameter tuning for the Random Forest model across the datasets did not result in significant performance improvements compared to the default settings. Therefore, to ensure consistency and allow for direct comparison of the modeling approaches across different datasets and feature sets, it was decided to use the default hyperparameter configurations provided by scikit-learn for Random Forest, AdaBoost, and Gradient Boosting as well as MLP throughout the experiments. However, on a final feature subset, a limited hyperparameter tuning was performed for all models. Table 3.4 shows the ranges of hyperparameters investigated, with their default values highlighted.

While other similar models, such as HistGradientBoostingRegressor and ExtraTreesRegressor, were also explored during initial testing, their results were found to be largely comparable in performance and behavior to the selected models. Thus, they were not included in the final reported comparisons for conciseness.

For deep learning, two architectures were selected based on recent publications demonstrating their effectiveness on the C-MAPSS dataset. The choice was guided by the availability of detailed architecture descriptions in the literature, which allowed for their reconstruction:

- **1D Convolutional Neural Network (1D-CNN):** A CNN architecture suitable for sequence data, adapted from the work of Ensarioğlu, İnkaya, and Emel [46]. Its specific layer configuration and hyperparameters are detailed in Table 3.3.
- **Hybrid CNN-GRU:** An architecture combining two convolutional layers for feature extraction from sequences with Gated Recurrent Units (GRU). This model was based on the description provided by Sun et al. [30], with its structure outlined in Table 3.2.

Table 3.2: CNN-GRU model architecture from Sun et al. [30].

Layer Type	Hyperparameters
Conv1D	filters = 128, kernel_size = 3, activation = relu
Dropout	rate = 0.1
MaxPooling1D	pool_size = 2
Conv1D	filters = 128, kernel_size = 3, activation = relu
Dropout	rate = 0.1
MaxPooling1D	pool_size = 2
GRU	units = 128, return_sequences = True
GRU	units = 128
Dense	units = 200, activation = relu
Dropout	rate = 0.2
Dense	units = 1, activation = linear

All the selected machine learning and deep learning models were systematically applied and evaluated across all three datasets (Primary Oil Filter Dataset, C-MAPSS Dataset, Dust Dataset) using the various feature sets developed during

Table 3.3: 1D-CNN model architecture from Ensarioğlu, İnkaya, and Emel [46].

Layer Type	Hyperparameters
Conv1D	filters = 32, kernel_size = 3, activation = relu
MaxPooling1D	pool_size = 3
Flatten	–
Dense	units = 50, activation = relu
Dense	units = 50, activation = relu
Dense	units = 1, activation = linear

the preprocessing and feature engineering phases (including raw/domain-specific features and the automatically generated features before and after PCA).

For the sequential deep learning model training, further hyperparameters needed to be chosen. The batch size was set according to the description in the literature to 128. The number of epochs was set to 30. To avoid overfitting, an early stopping callback was used with a patience of 5. The training loss was set to the mean squared error.

The sequential deep learning models receive a sequence of data for training. Using the full cycles as one sequence would dramatically limit the available training data due to the limited cycle count of the Primary Oil Filter Dataset. Instead, a fixed window approach was chosen, as proposed in original work by the authors [46, 30]. The window sizes were set according to the length of the individual cycles of the training data and therefore are not the same for all datasets.

For the C-MAPSS dataset, a window size of 30 was chosen as proposed by Ensarioğlu, İnkaya, and Emel [46] and Sun et al. [30]. For the Primary Oil Filter Dataset, the window size was set to 168, corresponding to a time window of 7 days. For the Dust Filter Dataset, the window size was set to 100.

3.4.3 Evaluation Strategy

The primary objective of the predictive modeling phase was to assess the performance and robustness of various machine learning and deep learning models for RUL estimation across different datasets and feature representations. A crucial aspect of designing the evaluation strategy was addressing the specific limitations of the Primary Oil Filter Dataset, particularly the extremely low number of available filter cycles.

A fundamental principle in model evaluation is to prevent information leakage by ensuring that data used for training a model is strictly separated from the data used for testing or validation. Given the time-series nature of the data and the focus on predicting the remaining life of a system, using segments from the same filter cycle in both the training and evaluation sets would lead to unrealistically optimistic performance estimates. However, with only four complete cycles available in the primary dataset, standard k-fold cross-validation approaches that randomly split data points across folds are unsuitable, as they could place data points from the

same cycle into different folds.

To address this critical challenge for the Primary Oil Filter Dataset, a Leave-One-Out Cross-Validation (LOOCV) approach was adopted to maintain the integrity of cycle information. Therefore, each of the four complete filter cycles serves as the dedicated test set exactly once. The model is trained on the data from the remaining three cycles and then evaluated on the left-out cycle. This process is repeated four times, ensuring that the model's performance is assessed on data from a cycle it has not seen during training, thus providing a more realistic estimate of its generalization capability to new, complete filter cycles.

To implement this cycle-based splitting strategy consistently, scikit-learn's GroupKFold cross-validation splitter was utilized. GroupKFold ensures that all data points belonging to the same "group" (in this case, a filter cycle) are kept together within the same fold. Although the complementary datasets have a larger number of cycles, a 4-fold GroupKFold setup was also applied, maintaining a consistent evaluation structure, even if each fold then contained multiple cycles rather than a single left-out one.

The primary metric used to evaluate model performance was the R^2 score. R^2 (coefficient of determination) measures the proportion of the variance in the dependent variable (RUL) that is predictable from the independent variables (features). An R^2 score of 1 indicates a perfect prediction, while a score of 0 indicates that the model performs no better than simply predicting the mean of the target variable. Negative R^2 scores indicate that the model performs worse than predicting the mean. The R^2 score was chosen as the main metric because it provides a standardized measure of predictability that allows for direct comparison of model performance across the inherently different scales and characteristics of the three datasets.

Furthermore, the effectiveness of a model can be effectively determined: An absolute minimum requirement for any useful prediction is an R^2 score greater than 0. For the Primary Oil Filter Dataset in particular, this minimum score is desired to be achieved when evaluated for all filter cycles. For example, if a model produces accurate predictions for three of the four cycles, but fails on the remaining cycle, this is a strong indicator that the models have not learned the complete underlying pattern of the filter degradation process.

Therefore, special attention was paid to the minimum performance (lowest R^2) achieved across the four folds in the cross-validation process of the Primary Oil Filter Dataset. This minimum score provides valuable insight into the consistency and reliability of the model's predictions when evaluated on individual filter cycles. For the supportive datasets, only the standard deviation of R^2 was inspected, since overall consistency is assumed to be significantly higher and cannot be traced back to individual cycles.

In addition to R^2, the Root Mean Squared Error (RMSE) was calculated for the supportive datasets. For the C-MAPSS dataset, RMSE, particularly for a low-RUL range (near end-of-life), was calculated to allow comparison with results from academic literature.

The evaluation strategy encompassed testing the selected machine learning and deep

learning models on different feature set variations developed during the preprocessing and feature engineering steps:

- The basic feature set derived from the raw sensor and laboratory data.
- The domain-specific extended feature set (including domain-specific features and metadata-related features).
- The feature set automatically generated using tsfresh (applied only to traditional ML including MLP).
- The dimensionality-reduced feature set after applying PCA to the tsfresh-generated features (applied only to traditional ML models).

A specific evaluation scenario for the Primary Oil Filter Dataset involved comparing model performance with and without the cumulative oil volume feature explicitly included. This test was designed to investigate the hypothesis, supported by exploratory data analysis insights, that this single feature might be overwhelmingly dominant or even necessary for achieving any meaningful predictions on the limited dataset.

Feature Importance

Feature importance mechanisms allow for an estimation of which features contribute to a model's predictions, and how important their contribution is. The machine learning library scikit-learn provides this functionality as a built-in method for its models.

For the tree-based machine learning models (Random Forest, AdaBoost, and Gradient Boosting), feature importance was analyzed on the Primary Oil Filter dataset. Feature importance scores were calculated using scikit-learn's built-in functions, which quantify the relative contribution of each feature to the model's prediction. To understand the overall significance of each feature and the consistency of its importance across different filter cycles, the mean and standard deviation of the feature importance scores were computed across the four folds of the Leave-One-Cycle-Out cross-validation. This analysis provided insights into which features were most consistently relevant for predicting RUL on this specific dataset.

Based on the findings from this feature importance analysis, a selected subset of features was curated. This subset aimed to include the features identified as most potentially relevant and consistently important for RUL prediction, while excluding features found to have negligible or unstable importance. This curated subset of features was then used for a final evaluation round to assess the performance of the selected models under potentially improved conditions.

For this final evaluation on the selected feature subset, a limited hyperparameter tuning was performed for both traditional machine learning models and deep learning models. This tuning focused on exploring a small range of values for a few key parameters for each model type, as detailed in Table 3.4. For the ensemble models (Random Forest, AdaBoost, Gradient Boosting), the primary parameter tuned was the number of estimators. For the MLP, both the number of hidden layers and the

number of neurons per layer were explored. For the deep learning models (CNN-GRU, 1D-CNN), the batch size was adjusted. This approach aimed to provide a first impression of how sensitive the models were to these core parameters and potentially achieve a basic improvement over default settings, rather than conducting an exhaustive hyperparameter optimization search.

Table 3.4: Hyperparameter for the tuning process, with the **default values** (used in the other phases) highlighted.

Model	Hyperparameter	Optimal Value
RandomForest	No. of estimators	30, 50, **100**, 300
AdaBoost	No. of estimators	30, **50**, 100, 300
Gradient Boosting	No. of estimators	30, 50, **100**, 300
MLP	Number of hidden layers	**1**, 2
MLP	Number of neurons per layer	25, 50, **100**, 200, 400
CNN-GRU	batch-size	32, 64, **128**, 256
1D-CNN	batch-size	32, 64, **128**, 256

By employing this systematic evaluation strategy, the project aimed to provide a comprehensive assessment of model performance, the utility of different feature engineering approaches, and the impact of dataset limitations for reliable RUL estimation.

3.5 Particle Sensor Study Methodology

The successful implementation of predictive maintenance for fluid systems relies on comprehensive oil condition monitoring. While this work's market study focuses specifically on inline particle sensors due to the identified gap in the primary dataset concerning frequent particle contamination data, it is important to acknowledge that a broader range of online sensor technologies exists for monitoring other critical oil properties.

For instance, several commercially available multi-parameter sensors utilize various physical sensing principles to simultaneously measure properties such as viscosity, density, and relative permittivity in-line [47]. Chemical degradation indicators, such as changes in acidity, basicity, and water content, can be targeted using sensor arrays. One approach involves capacitive sensors where each sensor in the array is coated with a specific absorbent material to enhance its sensitivity to target water or acidic/basic compounds [5]. Relative humidity sensors, often based on capacitive transducers, also provide an indirect indication of water content [47].

Commercially available fluid contamination sensors measure dielectric constant, conductivity, magnetic susceptibility, and micro-acoustic viscosity. Further optical contamination methods are based on transparency or refractometry and IR light absorption [48]. Beyond optical particle counters, sensors for detecting wear debris are based on magnetic methods [49].

The integration of multiple such online sensor technologies can provide a holistic view of oil degradation and system condition.

3.5.1 Purpose and Scope

The primary purpose of this market study was to investigate currently available online particle sensor technologies to identify feasible solutions for continuous monitoring of oil contamination. As highlighted in the analysis of the Primary Oil Filter Dataset, key indicators of oil degradation and filter clogging, such as particle count, MPC, and water content, were only available through infrequent laboratory measurements. Therefore, the goal of the study was to identify sensors capable of providing high-frequency particle contamination data that could be used alongside existing sensor data (temperature, pressure, flow, RH) and fill a critical gap in the data.

Multiple steps were conducted throughout this study. This involved consulting publicly available specifications and datasheets provided by manufacturers, searching vendor websites and technical literature. Search terms included ISO 4406 as an industry standard for particle contamination. Where necessary, suppliers and manufacturers were directly contacted to obtain additional technical information and inquire about price information. The study was conducted in close collaboration with operators from the hydraulic filtration systems, who provided valuable input regarding operational requirements. In total, the study investigated specifications for 23 different particle sensors from 14 distinct brands.

3.5.2 Key Performance Indicators (KPIs)

To evaluate the suitability of potential inline particle sensors for the target application, a set of Key Performance Indicators (KPIs) was defined. These KPIs were derived based on the operational characteristics of the hydraulic systems from which the primary dataset was collected, as well as the technical requirements for integration. These requirements were then discussed with the operators of the systems.

Operating Conditions Compatibility: Sensors must be capable of operating reliably under the specific temperature, pressure, and flow rate conditions encountered in the hydraulic presses. Based on the observed data from the Primary Oil Filter Dataset, the typical operating ranges were approximately 15 °C to 42 °C for temperature, 0.9 bar to 2.7 bar for pressure, and 0.3 L/min to 4.2 L/min for flow rate. Sensors were assessed on whether their specified operating ranges covered or exceeded these observed system conditions.

Particle Size Measurement Capability: A fundamental requirement was the ability to measure particle counts at sizes relevant to hydraulic fluid cleanliness standards. The minimum requirement was compatibility with the ISO 4406 standard particle size classes, which typically involve reporting counts for particles 4 µm, 6 µm, and 14 µm or larger. Sensors were evaluated on the specific particle sizes they could detect and report.

Accuracy: The accuracy of the particle count measurements is crucial for reliable

condition monitoring. While absolute accuracy can be challenging to compare directly across manufacturers, the study noted the reported accuracy specifications, often referenced against ISO standards or other calibration procedures.

Connectivity and Communication: For seamless integration into existing industrial control and data acquisition systems, the availability of standard industrial communication interfaces was a critical factor. Sensors were evaluated based on their support for protocols such as RS485, RS232, Modbus TCP/IP, CAN bus, or analog outputs like 4-20mA current loops.

Fluid Compatibility: Compatibility with the specific mineral oils used was essential and a mandatory requirement for all sensors. In addition, compatibility with water-based fluids was of interest to operators for potential future application expansion. Due to the specific requirements and specifications for synthetic oils, it was decided not to include supported synthetic oils in the comparison.

Price: While not a strict technical specification, the cost of the sensor is a practical consideration for feasibility and scalability of implementation across multiple machines. The study aimed to get an indication of the relative price of the different sensors.

These KPIs provided a structured framework for comparing the technical specifications and practical suitability of the investigated online particle sensors.

4

Results

4.1 Exploratory Data Analysis Insights

This section presents the findings from the exploratory data analysis (EDA) conducted on the Primary Oil Filter Dataset and the complementary datasets. The EDA aimed to uncover underlying patterns, relationships between variables, and potential challenges or indicators relevant to predictive modeling. Detailed descriptive statistics for the Primary Oil Filter Dataset are provided in Appendix A.1.

4.1.1 Primary Oil Filter Dataset

The Primary Oil Filter Dataset consists of four complete operational cycles. These cycles vary in duration: Cycle 1 lasted 195 days, Cycle 2 for 152 days, Cycle 3 for 200 days, and Cycle 4 was significantly shorter, lasting only 51 days.

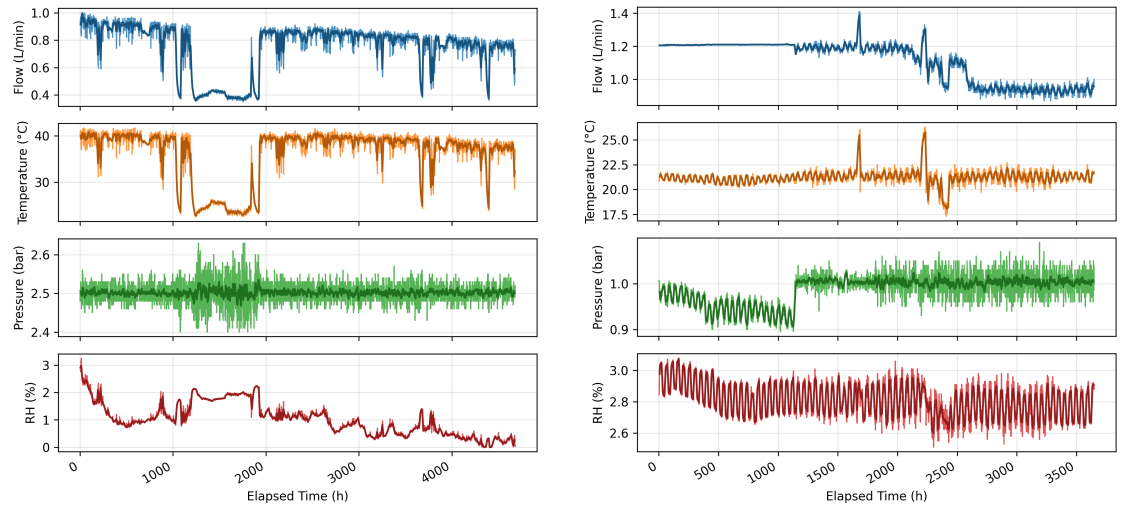
Visual inspection of the sensor data (Figure 4.1) reveals distinct characteristics for each cycle. Notably, the overall flow rate in Filter Cycle 4 is substantially higher, with an average flow approximately four times greater than in the other cycles. The pressure readings remain relatively consistent within each cycle, which is expected given the system's operational modes that target either specific pressure or flow values, as detailed in Section 3.3.1.

The laboratory measurements (Figure 4.2) also show inter-cycle variability. For instance, Filter Cycle 1 exhibits considerably lower particle counts across all measured size classes (4 µm, 6 µm, and 14 µm) compared to the other three cycles. The MPC values and water content also show different trends and magnitudes across the cycles.

A key observation from the historical trends is related to the cumulative oil volume. Figure 4.3 shows that despite the significant differences in cycle duration, all four filter cycles ended with a broadly similar total cumulative oil volume processed, ranging approximately between 214,000 and 274,000 liters. This suggests that the total oil throughput might be a more consistent indicator of filter end-of-life than operational time alone for this specific application.

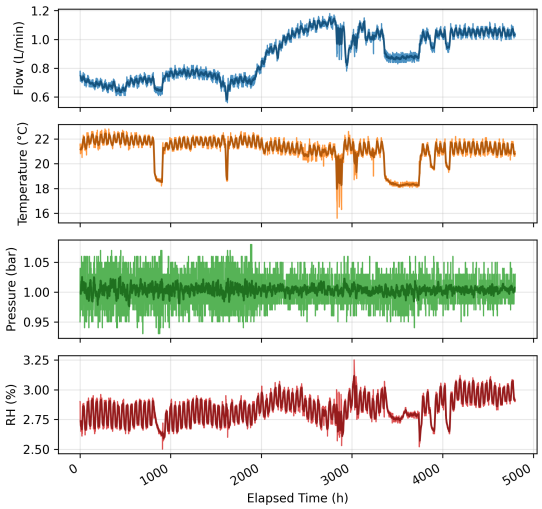
Time-related patterns

Further examination of the sensor data revealed patterns related to specific time periods. For example, significant drops in flow and temperature in Filter Cycles 1

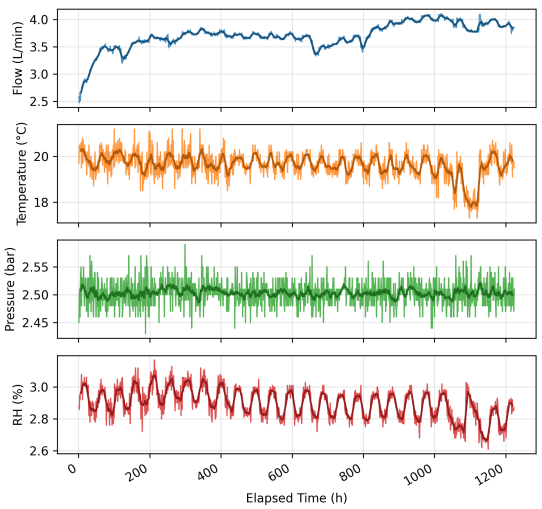


(a) Filter Cycle 1

(b) Filter Cycle 2



(c) Filter Cycle 3



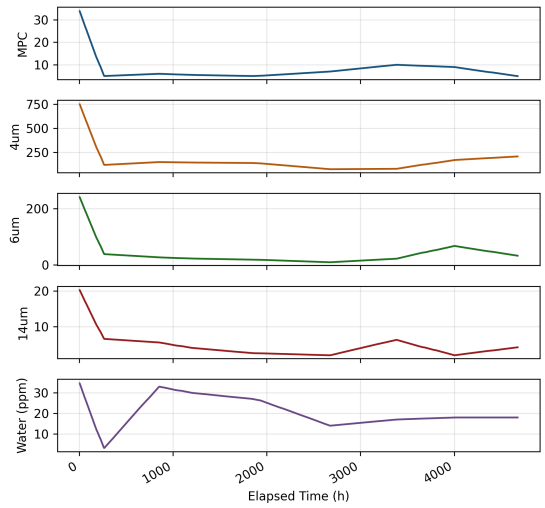
(d) Filter Cycle 4

Figure 4.1: Sensor data over the cycle time for all four cycles of the Primary Oil Filter Dataset. An enlarged version of the plot is presented in the Appendix A.2.

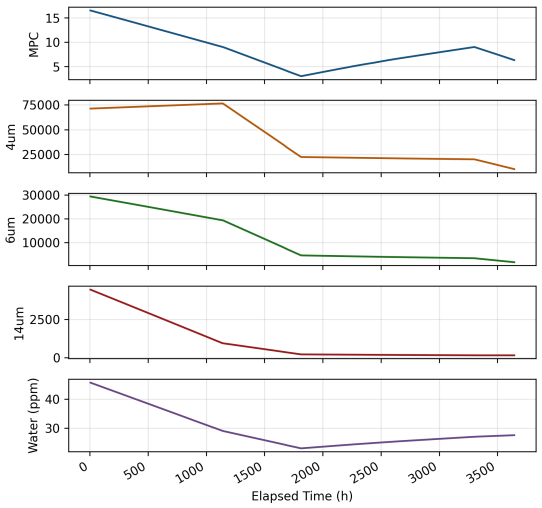
and 3 (visible in Figure 4.1) can be linked to external events such as the factory's summer break. Figure 4.4 illustrates the progression of temperature for Filter Cycle 3, overlaid with indications of public holidays in Sweden and the summer break period. These periods correspond to noticeable deviations in sensor readings, likely due to machine downtime or reduced operational load.

Daily and weekly periodicities were also observed in the data. Figure 4.5 displays heatmaps of temperature values averaged on a weekly time frame for Filter Cycles 1 and 3. These plots show that temperatures tend to be lower during weekends (Sunday, and to some extent Saturday) compared to weekdays, which is most dominant in these two cycles. Such patterns suggest that operational schedules influence the recorded oil parameters.

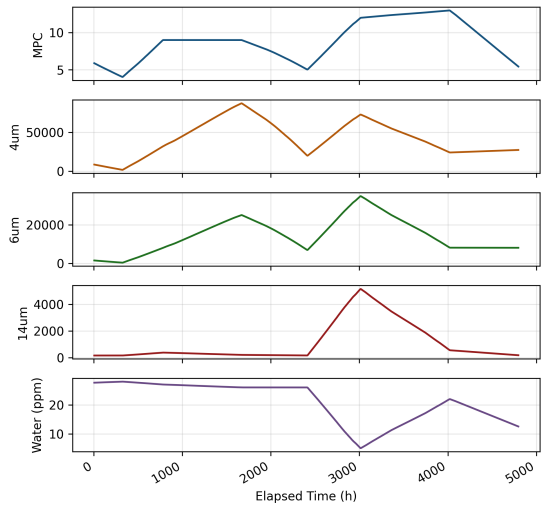
Relationships between sensor readings were explored using scatter plots. A notable



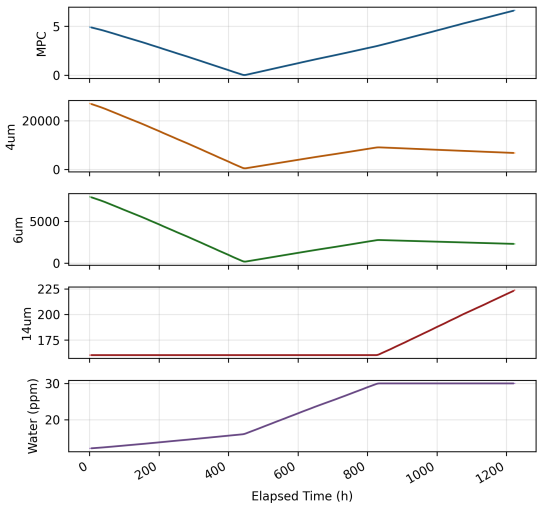
(a) Filter Cycle 1



(b) Filter Cycle 2



(c) Filter Cycle 3



(d) Filter Cycle 4

Figure 4.2: Laboratory measurement data over the cycle time for all four cycles of the Primary Oil Filter Dataset. An enlarged version of the plot is presented in the Appendix A.2.

relationship was found between Temperature and Relative Humidity (RH), as shown for Filter Cycle 1 in Figure 4.6 (left). In this plot, the color indicates the elapsed time within the cycle, allowing for an analysis of their co-dependent progression. The plot shows periods of slow temperature decrease followed by rapid increases. To investigate these rapid temperature changes, their first-order differences (derivatives) were calculated and plotted against RH, with the color dimension representing the hour of the day (Figure 4.6, right). Green dots, indicating high positive temperature derivatives, are frequently observed during morning hours. These findings suggest that sensor data, particularly temperature, is influenced by diurnal cycles or time-of-day related operational conditions.

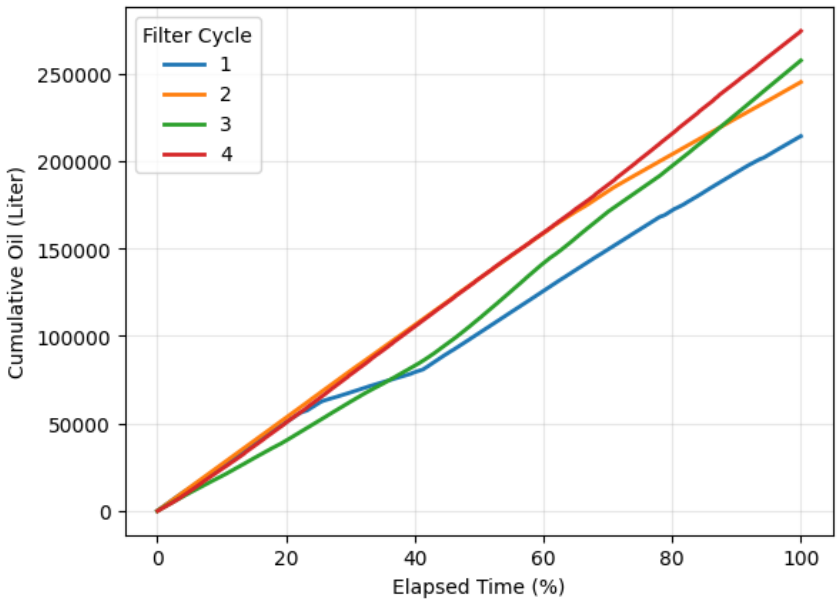


Figure 4.3: Cumulative oil volume for the filter cycles shown over the cycle time.

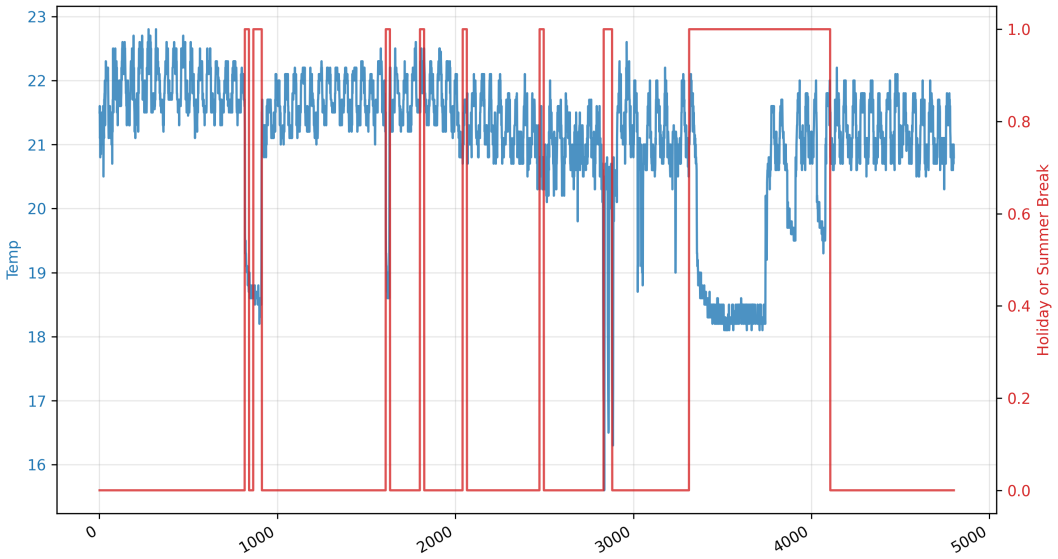
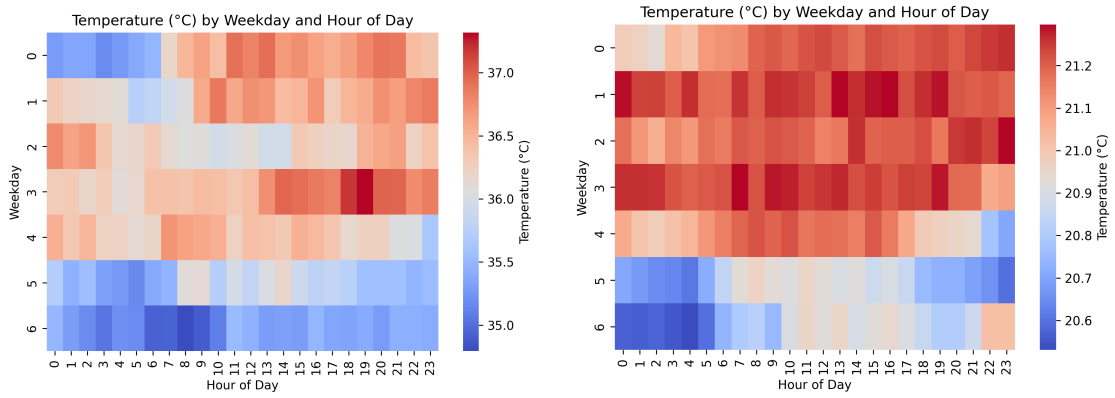


Figure 4.4: Progression of temperature together with indication of public holidays in Sweden, as well as the summer break period for Filter Cycle 3.

Correlation Analysis

Correlation analysis is a common tool for understanding relationships between variables and is often employed in RUL estimation problems, including those using the C-MAPSS dataset [30, 50]. It can indicate linear dependencies between features themselves and also help identify features potentially useful for prediction [51].

For datasets like C-MAPSS, a common approach is to treat the data as a whole when calculating correlations [9]. However, given the observed inter-cycle variability in the Primary Oil Filter Dataset (as seen in Figure 4.1), substantial deviations in



(a) Filter Cycle 1 **(b)** Filter Cycle 3

Figure 4.5: Heatmap of Temperature values averaged on a weekly time frame. Weekdays are numbered: Monday=0 to Sunday=6.

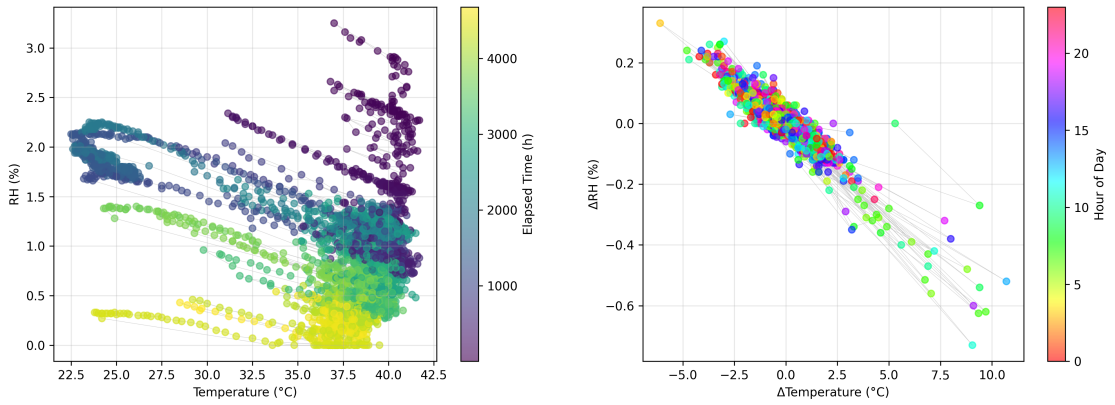


Figure 4.6: Scatter plot of temperature and RH values for Filter Cycle 1 (left), with color indicating elapsed cycle time. The corresponding first-order derivatives of temperature versus RH are shown on the right, with color indicating the hour of the day.

correlation structures between cycles were anticipated. Therefore, a more detailed analysis was conducted by calculating Pearson correlation coefficients for each of the four cycles independently. The average and standard deviation of these per-cycle coefficients were then evaluated.

Figure 4.7 presents the heatmap of average Pearson correlation coefficients, and Figure 4.8 shows the corresponding standard deviations. The most dominant average correlation is a strong negative relationship (coefficient of -1) between the time-based RUL target variable and the cumulative oil volume. The very low standard deviation for this correlation (Figure 4.8) indicates that this strong linear relationship is highly consistent across all four filter cycles. This reinforces the earlier observation about the potential importance of cumulative oil volume as a predictor.

Further prominent positive correlations are observed among the particle count mea-

surements and MPC values. For instance, particle counts for 6 µm and 4 µm show a high average positive correlation (coefficient of approximately 0.95), also with relatively low standard deviation, suggesting a consistent relationship.

In contrast, many other feature pairs exhibit high standard deviations in their correlation coefficients. For example, the average correlation between water content and time-based RUL is 0.21, but the standard deviation is a substantial 0.7. This high standard deviation indicates that the relationship between water content and RUL varies significantly from cycle to cycle; it might show a positive correlation in one cycle but a negative or no correlation in another. Such inconsistencies highlight the challenge of finding universally reliable predictors beyond cumulative oil volume from the available raw sensor and laboratory data.

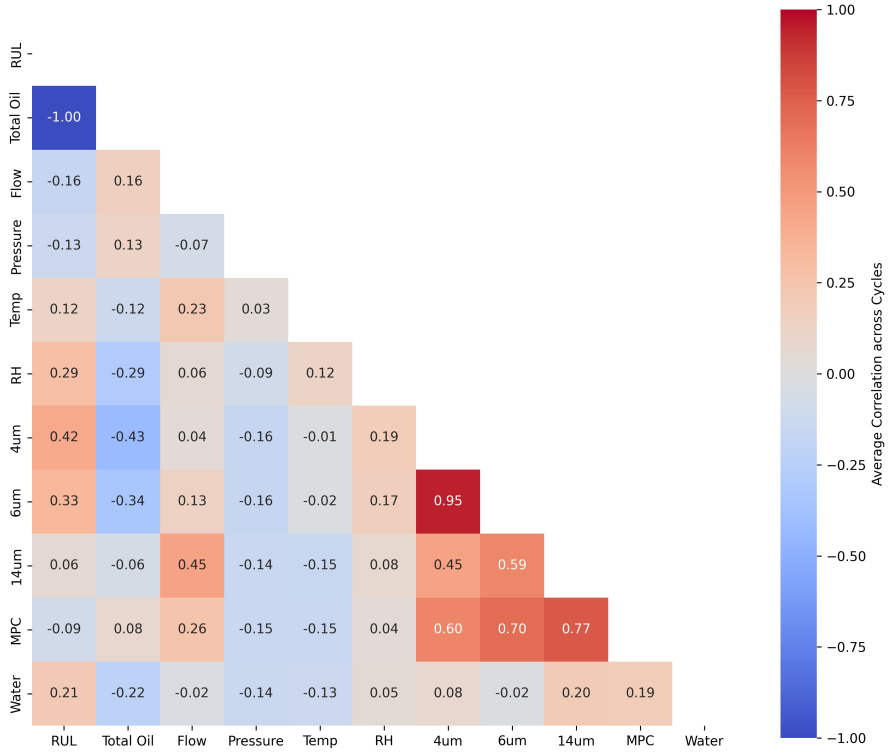


Figure 4.7: Heatmap of average pearson correlation coefficients for sensor and laboratory data of the Primary Oil Filter Dataset, calculated across the four individual cycles.

4.1.2 Complementary Datasets Insights

For comparative purposes, correlation analysis was also performed on the C-MAPSS and Dust Filter datasets, following a similar approach of calculating average cor-

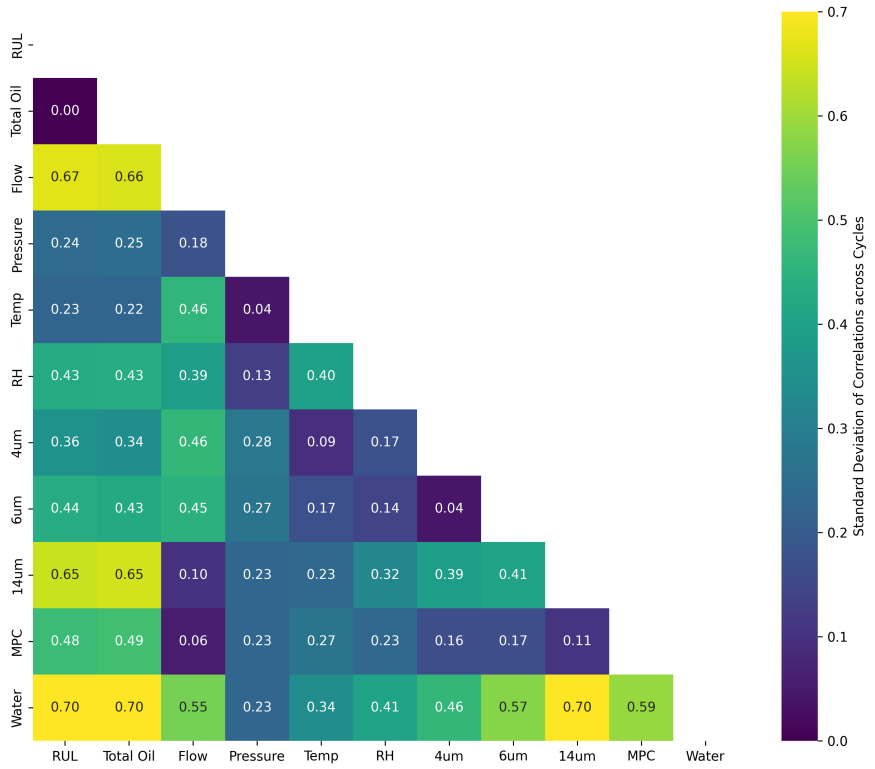
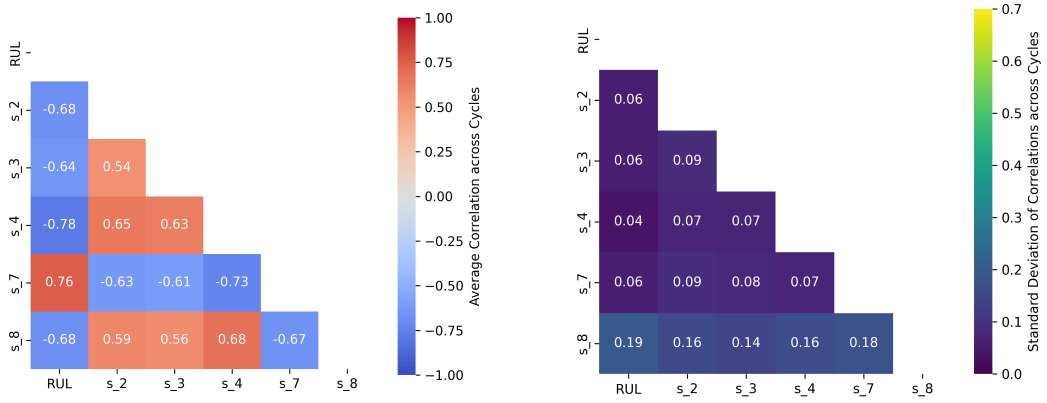


Figure 4.8: Heatmap of the standard deviation of pearson correlation coefficients for sensor and laboratory data of the Primary Oil Filter Dataset, calculated across the four individual cycles.

relations and their standard deviations across the individual units or cycles within each dataset.

For the C-MAPSS dataset (subset FD001), focusing on the selected 14 sensors, Figure 4.9 shows the average correlations and their standard deviations for a subset of these sensors with the RUL. Many sensors exhibit strong correlations with RUL, with absolute Pearson correlation coefficients often exceeding 0.5. Importantly, the standard deviations of these correlations across the 100 engine units are consistently small (typically below 0.2), indicating stable and reliable linear relationships between these sensors and RUL across the dataset.

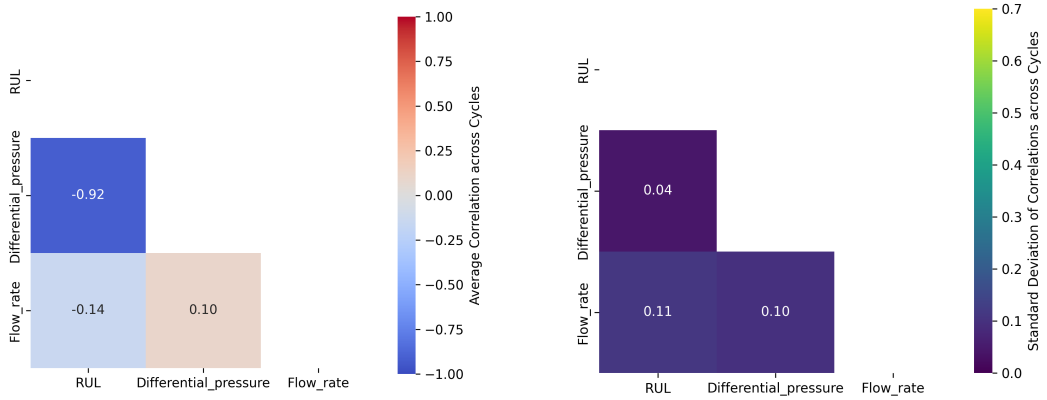
For the Dust Filter Dataset, the correlation analysis focused on a limited number of time-varying features, as others (like dust type) are constant within each run-to-failure trajectory. Figure 4.10 displays the average correlations and standard deviations. A dominant negative correlation coefficient of -0.92 is observed between RUL and differential pressure, with a relatively low standard deviation. This suggests that differential pressure is a strong and consistent linear indicator of RUL in this dataset. Other features, such as flow rate and dust feed rate, show much



(a) Average correlation. **(b)** Standard deviation of correlation.

Figure 4.9: Heatmaps of average pearson correlation coefficients (left) and their standard deviations (right) between selected sensors and RUL for the C-MAPSS Dataset. A subset of sensors is shown for conciseness; the full matrix is given in Appendix A.3.

smaller absolute correlation coefficients with RUL (below 0.2), indicating weaker linear relationships.



(a) Average correlation. **(b)** Standard deviation of correlation.

Figure 4.10: Heatmaps of average pearson correlation coefficients (left) and their standard deviations (right) between key sensor variables and RUL for the Dust Filter Dataset.

In summary, the correlation analyses on the complementary datasets reveal more consistent and often stronger linear relationships between sensor features and RUL compared to the Primary Oil Filter Dataset (excluding its cumulative oil volume feature). This highlights the more well-defined degradation signatures present in these larger, benchmark datasets, contrasting with the higher variability and complexity encountered in the primary dataset with its limited number of cycles.

4.2 Predictive Modeling Performance

This section presents the performance evaluation of the selected machine learning and deep learning models across the different datasets and feature sets. The results are analyzed and compared by model effectiveness and highlight the impact of data characteristics on predictive accuracy.

4.2.1 Primary Oil Filter Dataset

Given the results from the Exploratory Data Analysis, a clear indicator for the importance of the cumulative oil volume was found. For all filters, the total oil throughput at the end of the cycle was approximately the same. This insight was used to create a simple baseline model for reference. This model used linear regression with only one feature, the cumulative oil volume value.

The prediction across all four folds can be seen in Figure 4.11. Each fold represents the data of one filter cycle that was used for the validation of the model. In the plot, the predicted values of the validation data of each fold are plotted against the true values. For perfect prediction, all points would land on the diagonal. The individual filter cycles can still be identified as continuous lines. Observing the plot for the time-based RUL, it is noticeable that for one fold the predictions were significantly worse than for the remaining ones, with the remaining ones lying very close to the optimal prediction. The volume-based RUL on the other hand shows way more consistent predictions across all filter cycles.

The differences in prediction accuracy are evident in the R^2 scores. Table 4.1 shows that for the time-based RUL, one cycle has a negative R^2 score, whereas for the volume-based RUL all scores were positive.

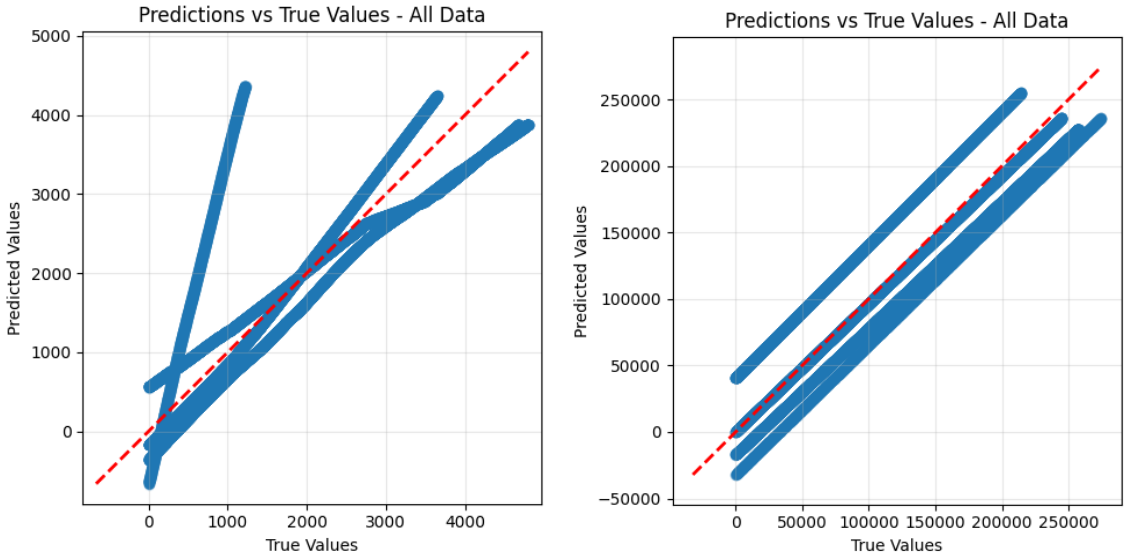
Table 4.1: R^2 scores for linear regression on cumulative oil volume. Minimum scores across the folds being underlined, and best scores are highlighted in **bold**.

RUL	Fold 1	Fold 2	Fold 3	Fold 4	Mean
Time	0.89	0.94	0.87	-22.60	-4.97 ± 11.75
Volume	**0.55**	1.00	0.90	0.80	**0.81** ± 0.19

Sensor and Laboratory Data

The investigated feature subset consists of the sensor and laboratory data. Predictions were made both including the cumulative oil volume, and excluding it.

Including the cumulative oil volume, the predictions are shown in Table 4.2. For the time-based RUL, all models achieve at least one negative score across the folds, similar to the linear model in Table 4.1. For fold 2 and fold 3, all models performed with a positive score value, whereas for fold 4, no model could achieve a positive score. The tree-based models scored well for three out of the four folds, but heavily failed on the last fold, resulting in large deviations in the R^2 score. The deep learning



(a) Time-based RUL **(b)** Volume-based RUL

Figure 4.11: Comparison of Linear Regression predictions for the two different RUL definitions.

models performed best on fold 2, with the CNN-GRU achieving the highest score of 0.94, resulting in the highest average score.

For the volume-based RUL, higher overall scores were accomplished by the tree-based models. Tree-based models achieved positive scores for all folds, while struggling with accurate predictions for fold 1. The best average score was achieved by AdaBoost, as well as the highest minimum score across the folds of 0.47. The MLP scored significantly worse compared to the time-based RUL with a negative score across all folds. Furthermore, none of the DL-models could achieve a positive score for all folds.

Excluding the cumulative oil volume, all models predicted significantly worse for both time-based and volume-based target values. The tree-based models and the MLP could not achieve positive R^2 scores. The 1D-CNN is the best scoring model in most folds. All scores are presented in Table 4.3.

Extended Feature Set

The extended feature set includes additional features, potentially including more information for the model. Complementing the sensor and laboratory data, this set includes the domain-specific derived features, as well as the metadata features. In total, 30 features (29 for excluding the cumulative oil count) were used for training and validation.

Table 4.4 shows the results of the prediction including the cumulative oil volume. For the time-based RUL, most models predicted worse compared to the original feature set for the majority of folds. The more complex CNN-GRU scored highest on average, still maintaining a negative score for fold 4. For the volume-based RUL,

Table 4.2: R^2 scores for sensor and laboratory data, **including** the cumulative oil volume.

Model	RUL	Fold 1	Fold 2	Fold 3	Fold 4	Mean
RandomForest	Time	0.44	0.68	0.71	-24.74	-5.73 ± 12.68
AdaBoost	Time	0.54	0.79	0.84	-24.23	-5.52 ± 12.48
GradientBoosting	Time	0.81	0.71	0.86	-20.81	-4.61 ± 10.80
MLP	Time	-2.05	0.83	0.93	-0.69	-0.24 ± 1.41
CNN-GRU	Time	0.76	0.94	0.57	-0.68	0.40 ± 0.73
1D-CNN	Time	-1.37	0.79	0.93	-3.68	-0.83 ± 2.17
RandomForest	Volume	0.34	0.91	0.83	0.73	0.70 ± 0.25
AdaBoost	Volume	**0.47**	0.96	0.89	0.74	**0.77** ± 0.22
GradientBoosting	Volume	0.22	0.95	0.91	0.70	0.69 ± 0.34
MLP	Volume	-3.09	-1.36	-2.48	-2.99	-2.48 ± 0.79
CNN-GRU	Volume	-0.03	0.26	-0.58	-0.04	-0.10 ± 0.35
1D-CNN	Volume	-2.00	0.92	0.95	0.89	0.19 ± 1.46

Table 4.3: R^2 scores for sensor and laboratory data, **excluding** the cumulative oil volume.

Model	RUL	Fold 1	Fold 2	Fold 3	Fold 4	Mean
RandomForest	Time	-2.61	-2.46	-2.17	-41.24	-12.12 ± 19.42
AdaBoost	Time	-2.89	-1.94	-1.42	-20.48	-6.68 ± 9.22
GradientBoosting	Time	-1.93	-2.18	-1.68	-21.10	-6.72 ± 9.59
MLP	Time	-2.39	-3.48	-0.49	-75.77	-20.53 ± 36.84
CNN-GRU	Time	-0.24	-0.00	-0.12	-0.71	-0.27 ± 0.31
1D-CNN	Time	0.09	-0.07	**-0.13**	0.73	0.15 ± 0.39
RandomForest	Volume	-3.96	-1.09	-0.46	-2.19	-1.93 ± 1.53
AdaBoost	Volume	-3.28	-0.96	-1.23	-1.52	-1.75 ± 1.05
GradientBoosting	Volume	-2.60	-1.24	-1.20	-1.75	-1.70 ± 0.65
MLP	Volume	-3.11	-1.17	-2.54	-3.00	-2.45 ± 0.89
CNN-GRU	Volume	-0.58	-0.03	-0.03	0.11	-0.13 ± 0.31
1D-CNN	Volume	-0.53	0.82	-0.13	0.47	**0.22** ± 0.58

average scores were similar or worse compared to the original set. The model with the highest average score of 0.70 and minimum score of 0.23 is AdaBoost.

Without the cumulative oil volume feature, predictions are significantly worse, with no model performing with positive scores across all folds. The resulting scores are shown in Table 4.5.

Table 4.4: R^2 scores for domain-specific extended feature set, **including** the cumulative oil volume.

Model	RUL	Fold 1	Fold 2	Fold 3	Fold 4	Mean
RandomForest	Time	-2.66	0.34	0.58	-33.18	-8.73 ± 16.37
AdaBoost	Time	-0.14	0.23	0.89	-35.19	-8.55 ± 17.77
GradientBoosting	Time	-0.21	0.52	0.90	-34.15	-8.24 ± 17.29
MLP	Time	-3.00	-1.69	-0.24	-0.06	-1.25 ± 1.38
CNN-GRU	Time	0.92	0.59	0.66	-0.67	0.38 ± 0.71
1D-CNN	Time	-1.76	-0.63	0.36	-0.46	-0.62 ± 0.88
RandomForest	Volume	0.05	0.89	0.62	0.23	0.45 ± 0.38
AdaBoost	Volume	**0.23**	0.95	0.75	0.88	**0.70** ± 0.32
GradientBoosting	Volume	0.07	0.97	0.85	0.92	**0.70** ± 0.42
MLP	Volume	-3.12	-2.09	-2.68	-2.93	-2.71 ± 0.45
CNN-GRU	Volume	-0.06	-0.04	-0.58	0.10	-0.15 ± 0.30
1D-CNN	Volume	-2.76	-0.15	0.36	0.46	-0.52 ± 1.52

Table 4.5: R^2 scores for domain-specific extended feature set, **excluding** the cumulative oil volume.

Model	RUL	Fold 1	Fold 2	Fold 3	Fold 4	Mean
RandomForest	Time	-2.37	-1.63	-1.45	-112.47	-29.48 ± 55.33
AdaBoost	Time	-2.44	-1.79	-0.91	-112.10	-29.31 ± 55.20
GradientBoosting	Time	-2.20	-2.19	-0.90	-103.62	-27.23 ± 50.93
MLP	Time	-3.33	-1.55	-0.33	-0.77	-1.50 ± 1.33
CNN-GRU	Time	0.87	-0.01	0.30	-0.68	**0.12** ± 0.65
1D-CNN	Time	-1.41	-0.46	-0.21	-2.42	-1.13 ± 1.01
RandomForest	Volume	-4.25	-0.79	0.44	-1.29	-1.47 ± 1.99
AdaBoost	Volume	-3.68	-1.06	-0.12	-1.10	-1.49 ± 1.53
GradientBoosting	Volume	-3.92	0.23	0.60	-1.38	-1.12 ± 2.06
MLP	Volume	-3.13	-1.95	-2.81	-3.05	-2.73 ± 0.54
CNN-GRU	Volume	**-0.59**	-0.00	-0.02	-0.03	-0.16 ± 0.29
1D-CNN	Volume	-2.86	-0.34	-0.10	-0.29	-0.90 ± 1.31

Automatically Extracted Features and PCA

This feature set consists of even more features than the previous extended feature set. Table 4.6 shows the resulting scores of the tested models. The deep learning models were not tested on this subset. The total oil volume is not specifically included as a feature, but indirectly through the cumulative summation method in the feature extraction algorithm.

The time-based RUL prediction scores are negative across all folds. For the volume-

based target RUL, AdaBoost and GradientBoosting score positive across the folds, with a minimum score of 0.12.

Table 4.6: R^2 scores for feature set generated with tsfresh.

Model	RUL	Fold 1	Fold 2	Fold 3	Fold 4	Mean
RandomForest	Time	-0.49	-0.78	0.49	-46.58	-11.84 ± 23.17
AdaBoost	Time	-0.76	-0.69	0.61	-39.12	-9.99 ± 19.43
GradientBoosting	Time	0.28	0.67	0.37	-40.21	-9.72 ± 20.32
MLP	Time	-4.08	-1.68	-1.79	-8.37	-3.98 ± 3.13
RandomForest	Volume	0.07	0.48	-0.15	-0.69	-0.07 ± 0.49
AdaBoost	Volume	0.39	0.67	0.41	**0.12**	0.40 ± 0.22
GradientBoosting	Volume	0.03	0.77	0.52	0.50	**0.46** ± 0.31
MLP	Volume	-3.17	-0.98	-2.00	-3.13	-2.32 ± 1.04

Table 4.7 shows the scores after applying the PCA dimensionality reduction to the feature set. The table shows that no tested model could achieve positive scores across the folds, with all average scores being negative.

Table 4.7: R^2 scores for PCA reduced tsfresh feature set.

Model	RUL	Fold 1	Fold 2	Fold 3	Fold 4	Mean
RandomForest	Time	0.35	-1.11	0.34	-6.03	-1.61 ± 3.03
AdaBoost	Time	-0.03	-1.94	0.10	-6.18	-2.01 ± 2.93
GradientBoosting	Time	0.29	-1.20	0.19	-12.83	-3.39 ± 6.33
MLP	Time	-2.54	-3.70	-3.02	-1.65	-2.73 ± 0.86
RandomForest	Volume	-2.48	-0.72	0.40	-1.14	-0.98 ± 1.19
AdaBoost	Volume	**-1.30**	-0.32	0.14	-0.48	**-0.49** ± 0.60
GradientBoosting	Volume	-1.60	-0.89	0.43	-0.20	-0.57 ± 0.87
MLP	Volume	-3.01	-2.60	-3.17	-3.08	-2.96 ± 0.25

4.2.2 Relevance of Features

For the tested subsets, the feature relevance was investigated. From the previous results on the dataset, it was found that predictions based on the volume RUL were significantly more accurate than prediction on the time-based RUL. In fact, none of the models could achieve positive R^2 scores across all folds for this scenario.

Furthermore, from the above tables we can see that the presence of the cumulative oil volume value is critical for any prediction by the models. Without this explicit feature, none of the models could predict the RUL consistently across the folds with positive R^2 scores, regardless of the feature set.

Therefore, the feature importance was analyzed only for the volume-based RUL including the cumulative oil volume.

For the first feature set, the feature importance of the tree-based models was similar: As expected, the highest importance across these models was the cumulative oil volume. The following features' importance values are significantly smaller. The relative feature importance of the best performing model is shown in Figure 4.12 for the five most dominant features.

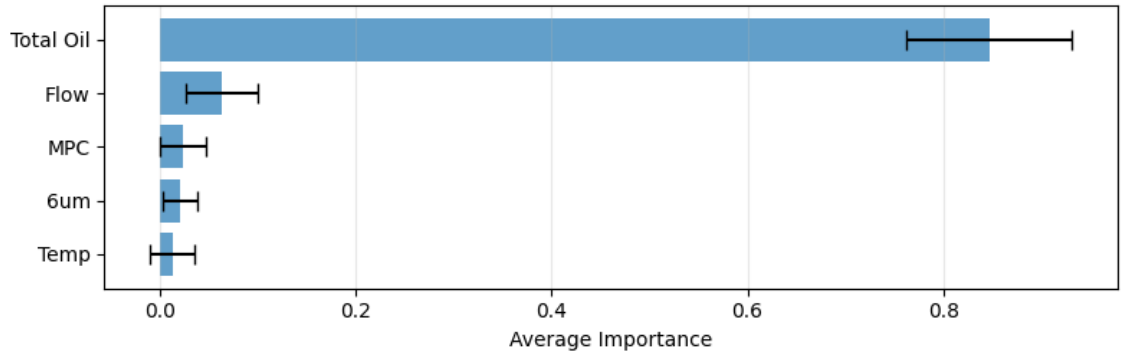


Figure 4.12: Relative feature importance for AdaBoost regressor on the sensor and laboratory data.

Investigating the feature importance of the domain-expanded feature set, we observe a similar contribution. However, the second most important feature is now the cumulative MPC value for all tree-based models, with a substantial distance to the following features. The following features include particle ratio, MPC value, and the absolute particle count value. The importance is plotted in Figure 4.13 for the best performing model.

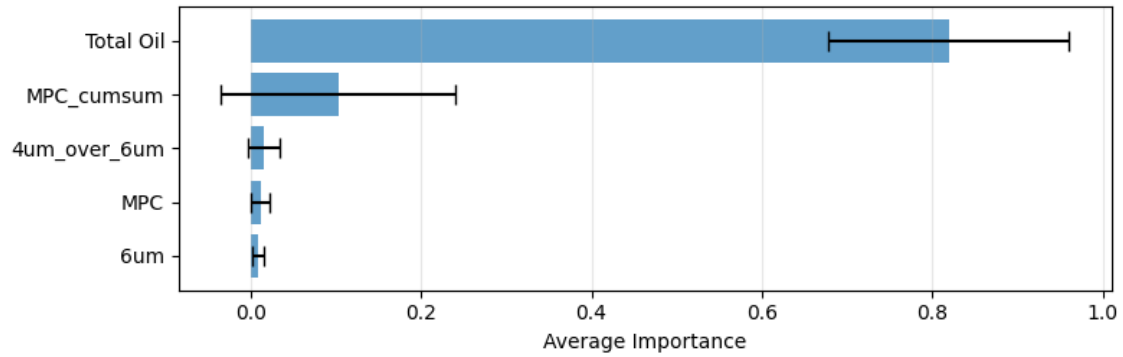


Figure 4.13: Relative feature importance for AdaBoost regressor on the domain-extended dataset.

The importance of the automatically generated features for the AdaBoost regressor is shown in Figure 4.14. This model was found to have the highest minimum R^2 value. Since the algorithm derived the cumulative sums of all features, the cumulative oil volume is now represented by the expanding window flow-summation feature. However, we can observe large standard deviations in the feature importance, not

only for the less important features, but also for the cumulative oil volume. The second most important feature, the difference of the setpoint running mode value from the flow (see Section 3.3.1), is also directly related to the cumulative oil volume.

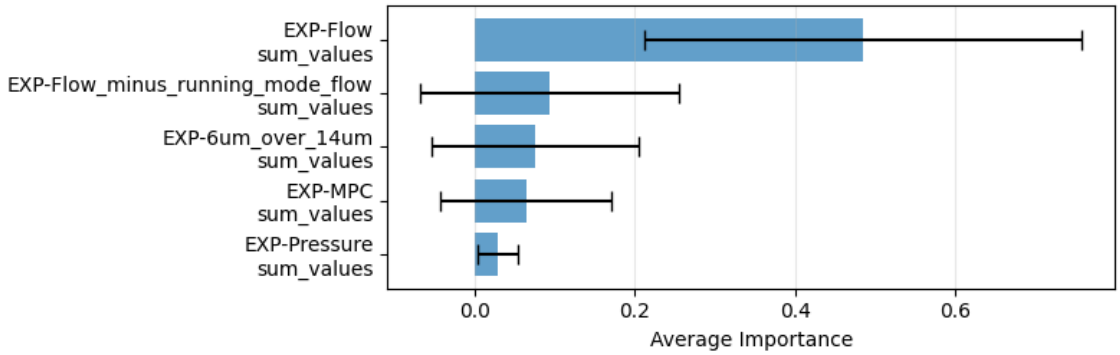


Figure 4.14: Relative feature importance for AdaBoost regressor on automatically generated features.

4.2.3 Evaluation on Relevant Features

The feature relevance analysis gives us some information on which features should be included for a well-performing model. Including more features did not necessarily increase the predictive performance. To again test the performance of the models on a subset of features, the results from the previous analysis were considered. The cumulative oil volume was found necessary for reliable prediction, and therefore was included in this feature subset. A further, but way less important feature, consistently showing up as the second-ranked important feature in the extended subset, is the cumulative MPC value. Laboratory measurement-derived features including MPC value and particle counts were found to rank in some upcoming positions. The RUL was defined as volume-based, since higher scores were achieved in the previous experiments.

For this reason, the final subset was decided to consist of these features:

- Cumulative oil volume
- Cumulative MPC value
- MPC value
- Particle count for 4 µm
- Particle count for 6 µm
- Particle count for 14 µm

Hyperparameter tuning was performed for all five models. For each model, the configuration achieving the highest minimum R^2 score across the folds was chosen. The best configuration found is given in Table 4.8.

The resulting R^2 scores are presented in Table 4.9. The best performing model in overall folds (0.89) as well as the highest minimum R^2 score (0.79) was found to be the optimized MLP model. The second largest minimum R^2 score was achieved

Table 4.8: Optimal hyperparameter configurations for the evaluated models.

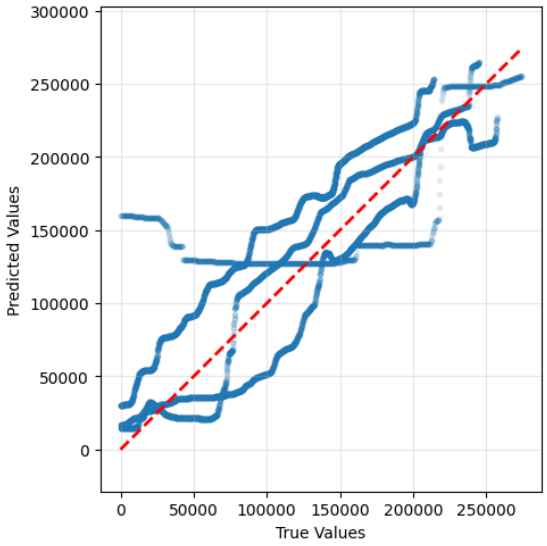
Model	Hyperparameter	Optimal Value
RandomForest	No. of estimators	300
AdaBoost	No. of estimators	300
Gradient Boosting	No. of estimators	300
MLP	Hidden layers	(200, 200)
CNN-GRU	batch-size	32
1D-CNN	batch-size	256

by the 1D-CNN model, scoring 0.72. The best performing tree-based model was GradientBoosting, scoring 0.85 on average over three of the subsets, and 0.59 on the remaining fold.

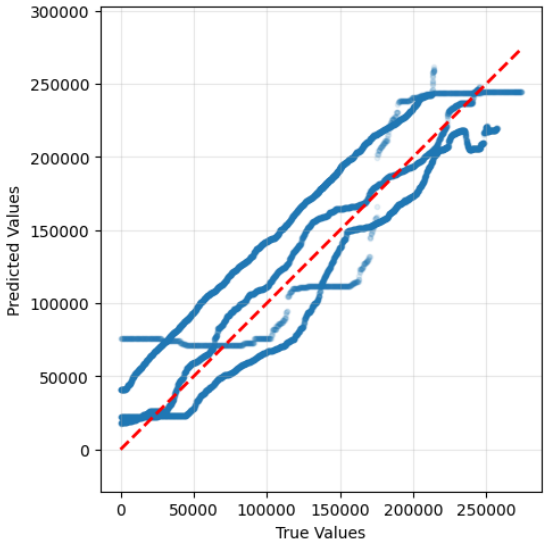
The predictions plotted against the corresponding true RUL values are shown in Figure 4.15. Even though the GradientBoosting and AdaBoost regressors appear consistent overall, some values show constant offset from the optimal prediction line. For MLP and 1D-CNN, most predictions are close to the optimum, while there seems to occur a spike in prediction for fold 1, resulting in large deviations. Since this applies for a few data points only, the overall score can still be higher compared to the more consistent tree-based models.

Table 4.9: R^2 scores for feature importance selected feature set. Simple hyperparameter tuning was performed.

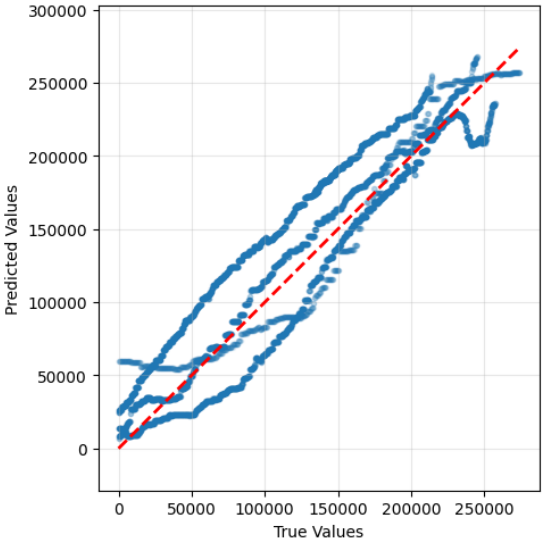
Model	Fold 1	Fold 2	Fold 3	Fold 4	Mean
GradientBoosting	0.59	0.98	0.91	0.92	0.85 ± 0.18
AdaBoost	0.53	0.97	0.90	0.84	0.81 ± 0.19
RandomForest	0.54	0.93	0.88	0.37	0.68 ± 0.27
MLP	0.83	0.96	**0.79**	0.96	**0.89** ± 0.09
CNN-GRU	0.75	0.87	0.84	0.46	0.73 ± 0.19
1D-CNN	0.84	0.79	0.82	0.72	0.79 ± 0.05



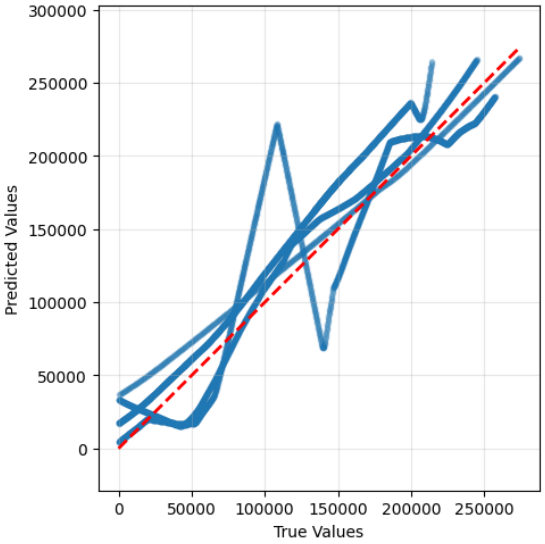
(a) RandomForest



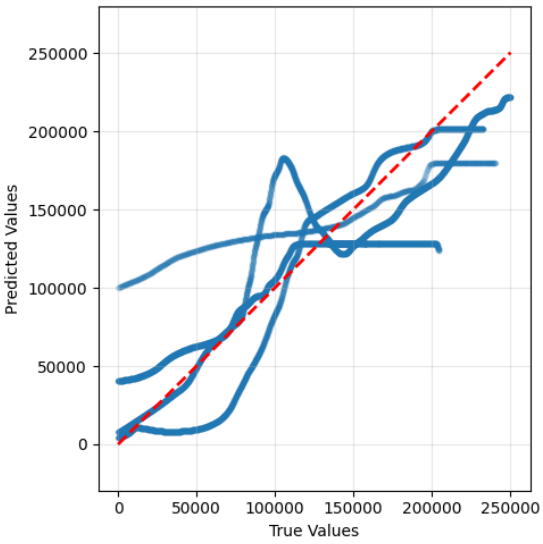
(b) AdaBoost



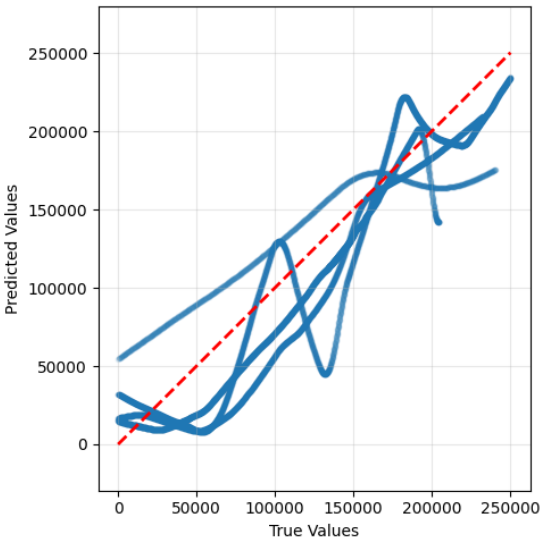
(c) GradientBoosting



(d) MLP



(e) CNN-GRU



(f) 1D-CNN

Figure 4.15: RUL prediction for the feature importance selected set.

4.2.4 Complementary Datasets

C-MAPSS dataset

The C-MAPSS Aircraft Engine Simulator Data was utilized as a complementary dataset. Given the slightly different nature of this data, the data is assumed to perform differently. The main focus of these experiments lies on the linearly defined RUL; however, knowing that the performance differs across the RUL values, a low-RUL threshold of 100 was set to investigate the prediction differences for the lower target values.

Original Sensor Set

To make the results more comparable to publicly available studies of the same dataset, a common approach for the target RUL definition was tested alongside the primary linear definition. For the results in Table 4.10, the target RUL was defined as linear decaying up to a value of 130. From this point, a constant RUL value of 130 was set, resulting in a piecewise linear target value.

The standard deviation explains the differences across the four different folds in the validation process. Results are consistent with a standard deviation up to 0.05 for the R^2 score. The best performing model is the 1D-CNN with an R^2 score of 0.85, closely followed by the CNN-GRU. MLP scored third best, followed by the tree-based models. RMSE values range from 16.53 for the 1D-CNN to 25.43 for the AdaBoost model. The errors in the low-RUL spectrum are similar to the errors across the full range.

These results can be compared with the resulting scores from the literature. However, due to changes in the selected features, training and evaluation setup, interpretation should be done with caution. In the paper from Sun et al. [30], the CNN-GRU model achieved an R^2 score of 0.91 with a corresponding RMSE of 13.48. These results are slightly better compared to the results from Table 4.10. For the 1D CNN model, Ensarioğlu, İnkaya, and Emel [46] reports an RMSE between 15.5 and 17.2 depending on the feature processing method. With an RMSE of 16.53, the results from Table 4.10 are right in between.

Table 4.10: Comparison of RUL prediction on the piecewise linear target RUL for the C-MAPSS dataset.

Model	R^2	RMSE	Low-RUL RMSE
RandomForest	0.78 ± 0.04	20.10 ± 1.77	22.46 ± 2.09
AdaBoost	0.66 ± 0.02	25.43 ± 0.71	21.39 ± 1.21
GradientBoosting	0.79 ± 0.03	20.04 ± 1.62	22.96 ± 2.01
MLP	0.80 ± 0.04	19.54 ± 1.85	22.29 ± 2.08
CNN-GRU	0.84 ± 0.04	17.14 ± 1.94	17.31 ± 2.59
1D-CNN	$\mathbf{0.85} \pm 0.02$	$\mathbf{16.53} \pm 1.14$	$\mathbf{16.18} \pm 0.99$

Table 4.11: Comparison of RUL prediction with linear target RUL definition for the C-MAPSS dataset.

Model	R^2	RMSE	Low-RUL RMSE
RandomForest	0.59 ± 0.07	43.76 ± 3.38	34.30 ± 3.47
AdaBoost	0.48 ± 0.09	49.57 ± 3.66	50.60 ± 4.46
GradientBoosting	0.60 ± 0.06	43.51 ± 3.22	33.84 ± 3.32
MLP	0.62 ± 0.06	42.80 ± 3.32	32.95 ± 3.38
CNN-GRU	0.66 ± 0.06	36.22 ± 2.71	27.81 ± 5.40
1D-CNN	$\mathbf{0.71} \pm 0.04$	$\mathbf{33.36} \pm 2.24$	$\mathbf{22.30} \pm 1.56$

Given the primary, completely linear RUL definition, the prediction results for the six investigated models are shown in Table 4.11. The best performing model is the 1D-CNN with an R^2 score of 0.71 and a corresponding RMSE of 33.36. The CNN-GRU model scored second with an average R^2 of 0.66 and RMSE of 36.22. MLP outperformed all tree-based models, with the lowest R^2 score for AdaBoost of 0.48. For the better-performing models, a significant difference in overall RMSE to low-RUL RMSE can be observed: The 1D-CNN achieved a value of 22.3 for the low-RUL range, decreasing the RMSE by 13.92. For other models, the difference is smaller, but still significant.

The actual predictions across the validation data are shown in Figure 4.16. For low true RUL values, the predicted correspondences are closer to the ideal line, indicating higher accuracy. For increasing RUL, the predictions get more spread out, indicating larger errors. For RUL values larger than 200, none of the models can predict the correct target value.

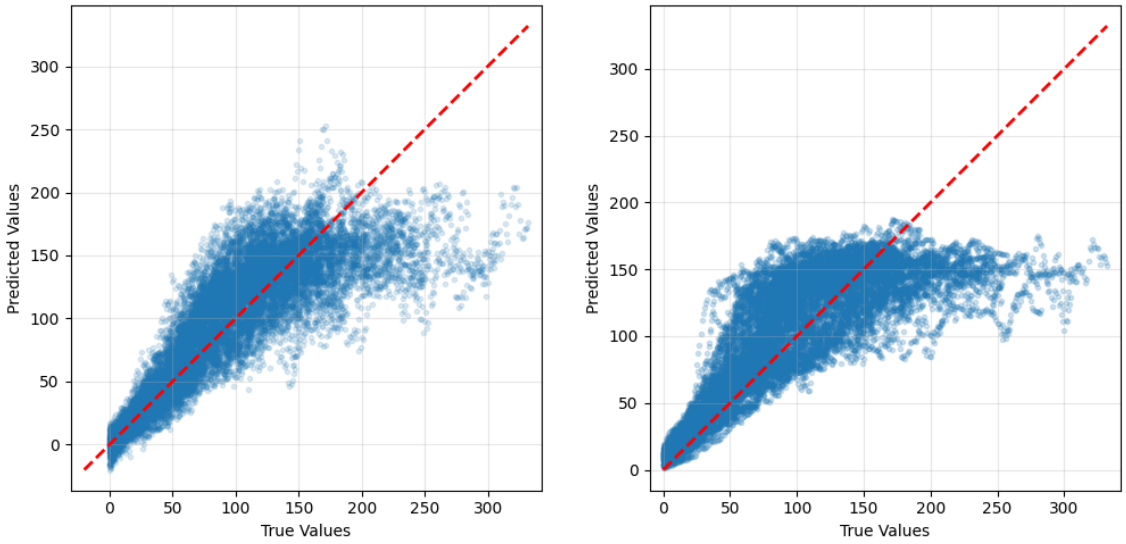
Automatically Extracted Features and PCA

The C-MAPSS Dataset was extended by the same feature creation method as the Primary Oil Filter Dataset. This significantly larger dataset was tested with the same models.

Table 4.12: Comparison of RUL prediction with for automatically extracted features for the C-MAPSS dataset.

Model	R^2	RMSE	Low-RUL RMSE
RandomForest	$\mathbf{0.70} \pm 0.03$	$\mathbf{34.18} \pm 1.97$	$\mathbf{18.46} \pm 3.73$
AdaBoost	0.67 ± 0.05	35.71 ± 2.74	29.04 ± 2.61
GradientBoosting	0.68 ± 0.04	34.76 ± 2.47	21.77 ± 2.08
MLP	0.56 ± 0.06	40.90 ± 2.44	26.98 ± 1.35

Table 4.12 shows the RUL predictions on the automatically extended dataset for the four machine learning models. The best performing model is RandomForest,



(a) 1D-CNN **(b)** GRU-CNN

Figure 4.16: Comparison of the two best performing models on the default sensor set of the C-MAPSS dataset.

with an R^2 of 0.70 and RMSE of 34.18. For the low-RUL range, the errors are even smaller, with an average of 18.46 across the folds. Other models scored with an R^2 of 0.68 (GradientBoosting) and 0.67 (AdaBoost).

Overall, we observe improved RUL prediction for all tree-based models compared to the original dataset without additional features. The RandomForest model score for the extended dataset is very close to that of the best performing model (1D-CNN) for the original set. Low-RUL predictions of RandomForest and GradientBoosting show lower errors than any other model on the original dataset.

Table 4.13: Comparison of RUL prediction with for automatically extracted features with PCA reduction for the C-MAPSS dataset.

Model	R^2	RMSE	Low-RUL RMSE
RandomForest	0.61 ± 0.06	38.70 ± 3.22	23.55 ± 4.73
AdaBoost	0.55 ± 0.07	41.56 ± 2.66	38.46 ± 2.97
GradientBoosting	0.68 ± 0.06	35.19 ± 3.09	24.35 ± 3.15
MLP	$\mathbf{0.70} \pm 0.04$	$\mathbf{33.63} \pm 2.39$	$\mathbf{21.40} \pm 1.44$

Subsequently, PCA was applied to the set of automatically extracted features to reduce dimensionality. The performance of the models on this PCA-reduced feature set is detailed in Table 4.13. With the PCA-transformed features, the MLP model emerged as the top performer, achieving an R^2 score of 0.70 and an RMSE of 33.63. This model also yielded the lowest RMSE of 21.40 in the low-RUL range. GradientBoosting followed with an R^2 of 0.68. Comparing these results to the non-PCA transformed extended feature set (Table 4.12), PCA application led to a slight decrease in performance for most tree-based models; for instance, the RandomForest

R^2 score dropped from 0.70 to 0.61. However, PCA notably improved the MLP's performance, with its R^2 score increasing from 0.56 to 0.70, allowing it to surpass the other models on this dataset.

Preventive to Predictive Maintenance Dataset

The Preventive to Predictive Maintenance dataset, focusing on dust filter degradation, was also subjected to the same modeling approaches.

Original Feature Set

The RUL prediction performance on the original feature set of the dust filter dataset is presented in Table 4.14. All evaluated models demonstrated strong predictive capabilities, achieving high R^2 scores. The 1D-CNN model emerged as the top performer, yielding an R^2 score of 0.95 and an RMSE of 156.10. The MLP also performed exceptionally well with an R^2 of 0.94 and an RMSE of 169.03, closely followed by CNN-GRU and GradientBoosting, both achieving R^2 scores of 0.93.

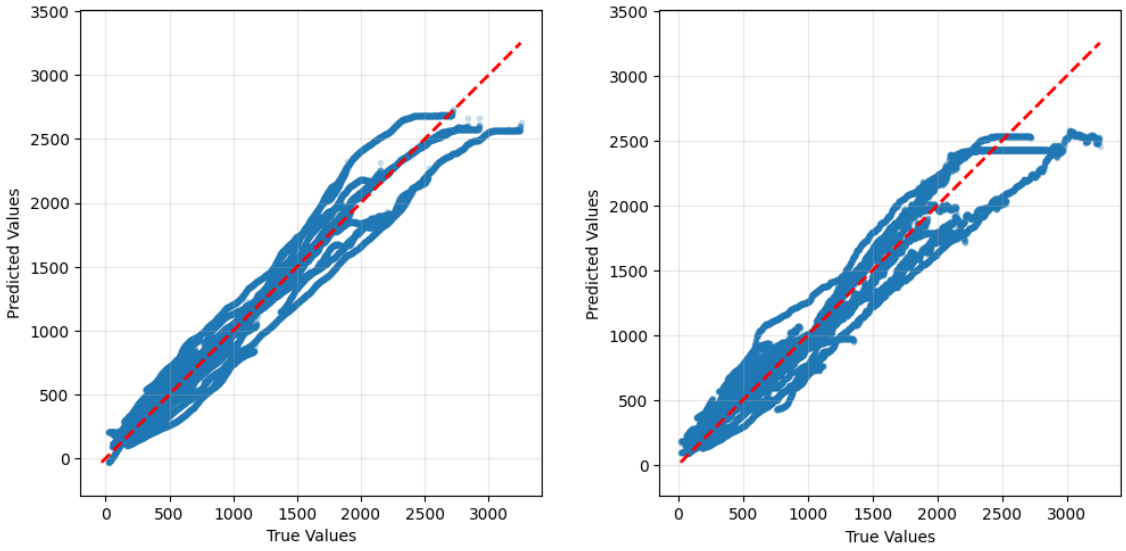
Table 4.14: Comparison of RUL prediction on for the original feature set for the dust filter dataset.

Model	R^2	RMSE
RandomForest	0.89 ± 0.05	233.38 ± 19.37
AdaBoost	0.92 ± 0.03	205.59 ± 49.61
GradientBoosting	0.93 ± 0.04	184.26 ± 40.95
MLP	0.94 ± 0.04	169.03 ± 56.87
CNN-GRU	0.93 ± 0.02	184.81 ± 39.02
1D-CNN	$\mathbf{0.95} \pm 0.03$	$\mathbf{156.10} \pm 54.68$

Figure 4.17 illustrates the predictions of the two best-performing deep learning models (1D-CNN and CNN-GRU) against the true RUL values. The plots show a good alignment between predicted and actual RUL, particularly for lower RUL values (<500), indicating accurate predictions as the filters approach their end-of-life. Given the reduced cycle count compared to the C-MAPSS Dataset, some individual lines appear in the plots, potentially corresponding to individual cycles. Even for large RUL values, for filters not close to their end-of-life, predictions appear bundled around the target line, without major deviations.

Automatically Extracted Features and PCA

The performance of the machine learning models on the automatically extended feature set, generated using tsfresh, is shown in Table 4.15. GradientBoosting was the best performing model on this feature set, with an R^2 score of 0.91 and an RMSE of 222.56. RandomForest also achieved a good R^2 of 0.89. Compared to the performance on the original feature set (Table 4.14), where deep learning models excelled, the tree-based models with extended features achieved slightly lower R^2



(a) 1D-CNN **(b)** GRU-CNN

Figure 4.17: Comparison of the two best performing deep learning models for the original sensor set for the dust filter dataset.

scores and higher RMSE values. For instance, GradientBoosting's R^2 of 0.91 on the extended set is slightly below its 0.93 on the original set, and its RMSE is higher.

Table 4.15: Comparison of RUL prediction on the automatically extended feature set for the dust filter dataset.

Model	R^2	RMSE
RandomForest	0.89 ± 0.07	228.64 ± 71.71
AdaBoost	0.86 ± 0.05	265.46 ± 44.59
GradientBoosting	$\mathbf{0.91} \pm 0.01$	$\mathbf{222.56} \pm 33.08$
MLP	0.86 ± 0.07	271.72 ± 87.96

Following feature extraction, PCA was applied for dimensionality reduction. The results for the PCA-reduced set are presented in Table 4.16. GradientBoosting maintained its position as the best model in this scenario, achieving an R^2 of 0.90 and an RMSE of 221.58. Other models, such as RandomForest and AdaBoost, showed a slight decrease in their R^2 scores compared to the full automated feature set, scoring 0.84 and 0.83 respectively. The MLP model's performance also declined on the PCA-reduced set compared to its strong performance on the original features. Overall, while PCA simplified the feature space, it generally led to a marginal reduction in predictive performance compared to using the full set of automatically extracted features, and a more noticeable gap when compared to the models' performance on the original, more compact-sized feature set.

Table 4.16: Comparison of RUL prediction on the PCA-reduced set for the dust filter dataset.

Model	R^2	RMSE
RandomForest	0.84 ± 0.04	286.28 ± 36.98
AdaBoost	0.83 ± 0.06	297.96 ± 54.19
GradientBoosting	$\mathbf{0.90} \pm 0.03$	$\mathbf{221.58} \pm 37.28$
MLP	0.82 ± 0.05	306.23 ± 70.51

4.3 Particle Sensor Study Findings

The market study on inline particle sensors provides a comprehensive overview of available technologies, detailing their capabilities and compatibility for enhancing the existing data acquisition system.

Evaluation of Suitable Sensors

This section summarizes the findings from the market study of inline particle sensors. The study aimed to identify sensors capable of providing continuous oil cleanliness data to complement existing measurements from hydraulic systems. A detailed comparison of the 23 investigated sensors across 14 brands is presented in Table 4.17.

It is important to note that not all manufacturer datasheets provided complete information for every Key Performance Indicator (KPI). Where specific data was unavailable, entries in the comparison table are marked as "N/A" or "-". Price information, when obtainable, was converted to a common currency (SEK) and categorized into ranges (€= up to 3000 SEK, €€= 3000-6000 SEK, €€€= over 6000 SEK). These price indications are approximate and may vary based on region, supplier, and specific quotations. For several sensors, price information was requested but not received within the study's time frame.

The investigated sensors offer a variety of capabilities. Particle size measurement typically covers the standard ISO 4406 channels (4 µm, 6 µm, 14 µm), with many sensors offering additional channels up to 70 µm or 100 µm, such as Pos. 5 and Pos. 8. Some specialized sensors, like Pos. 21 with optional extension, can detect particles as small as 0.5 µm. Accuracy is commonly reported as ±0.5 to ±1 ISO code.

Operating condition specifications vary widely. Maximum operating pressures often reach 420 bar (e.g., Pos. 1, Pos. 4), though some are designed for lower pressures like Pos. 6 at 50 bar. Flow rate capabilities are typically in the range of tens to hundreds of milliliters per minute, with some exceptions designed for higher flow, such as Pos. 6 allowing a flow rate up to 50 L/min. Temperature operating ranges are generally broad, commonly accommodating −20 °C to 85 °C (e.g., Pos. 1, Pos. 4).

A few sensors incorporate specific design features. For instance, Pos. 20 and Pos. 21 include integrated pumps, which control the flow rate through the sensor, at values such as 25 mL/min (for Pos. 20) or 10 mL/min to 25 mL/min (for Pos. 21). Sensors

Pos. 2, Pos. 3, Pos. 9, Pos. 10, and Pos. 11 utilize digital imaging technology. This often allows for advanced features such as particle shape recognition, differentiation between solid particles and air bubbles, and, in some cases, oil degradation analysis. The majority of other sensors are based on the light extinction or light obscuration principle (terms are used interchangeably in the datasheet), although this is not explicitly stated in all documents.

Several products across the specified price ranges are available. Sensors with an extended particle size range can also be found in the lowest price range (Pos. 8, Pos. 16). Digital imaging technology is found in the middle and higher price range.

Fluid compatibility is generally broad for mineral oils, which was a primary requirement. Some sensors, such as Pos. 2, Pos. 3, Pos. 11, Pos. 16 (N version), Pos. 21, and Pos. 23, also list compatibility with water or water-based fluids, which was noted as an interest for future applications. While not a primary focus of this comparison, many sensors also support various synthetic oils; for instance, Pos. 1 and Pos. 4 list compatibility with synthetic esters and PAOs. For specific or aggressive fluid compatibility, direct consultation with the supplier is always recommended.

Industrial communication protocols are widely supported. Common interfaces include RS485 (often with Modbus RTU), CAN bus (including CANopen or J1939 variants), RS232, and Modbus TCP/IP via Ethernet. Analog output, typically a 4-20mA current loop, is also a prevalent option, supported by at least 17 of the 23 sensors.

Compatibility with Sensor Ranges

The study assessed sensor specifications against the typical operating conditions observed in the Primary Oil Filter Dataset: temperature 15 °C to 42 °C, pressure 0.9 bar to 2.7 bar, and flow rate 0.3 L/min to 4.2 L/min.

Regarding pressure, most sensors can handle the maximum system pressure. However, some sensors have minimum operating pressure requirements that exceed the lower end of the system's typical range (0.9 bar). Specifically, Pos. 7 with a minimum of 1.5 bar, Pos. 17 with a minimum of 25 bar, Pos. 18 with a minimum of 2 bar, and Pos. 22 with a minimum of 2 bar might not be suitable for continuous operation if system pressure drops to the lowest observed values.

For direct in-line placement matching the full system flow rate (0.3 L/min to 4.2 L/min, or 300 mL/min to 4200 mL/min), most of the investigated sensors are unsuitable as their maximum supported flow rates are considerably lower (e.g., typically up to 400 mL/min or 500 mL/min). Pos. 6, with a flow up to 50 L/min, is an exception that can accommodate the higher flow rates of the system. However, its minimum flow requirement of 0.5 L/min is slightly above the system's observed minimum of 0.3 L/min, which might be problematic.

The operating temperature ranges of most sensors are compatible with the system's typical temperature range; only Pos. 21 does not specifically mention operation temperatures.

Filter Bypass

Given the flow rate limitations of most particle sensors relative to the main hydraulic system flow, in-line placement of the sensor can be problematic. A common solution is to install the sensor in a filter bypass loop, as illustrated in Figure 4.18. This configuration allows a small, controlled portion of the oil to be diverted through the sensor.

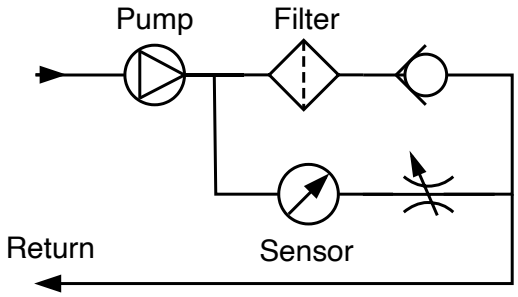


Figure 4.18: Schematic drawing of bypass solution for particle sensing in the filter setup. The sensor is located on-line. Additional valves allow to control and constrain the flow.

Creating a bypass offers several advantages. The flow through the sensor becomes independent of the main system flow, allowing it to be regulated to the optimal rate specified by the sensor manufacturer for maximum accuracy and reliability. This ensures consistent measurement conditions irrespective of variations in the main circuit.

However, a consideration with a bypass arrangement is that a small volume of oil continuously circulates through the sensor loop without passing through the main system filter. While typically a very small fraction of the total oil volume, this means this portion of oil is not being actively cleaned by the filtration system during its passage through the sensor loop. The impact of this is generally considered negligible in systems with adequate overall filtration and mixing.

Table 4.17: Product comparison for in-line particle sensors, showing product name, brand, supported particle sizes, accuracy, operating conditions and price information.

Pos	Product Name	Brand	Particle Size (µm)	Accuracy (ISO)	Pressure (bar)	Flow ($mL\,min^{-1}$)	Temperature (°C)	Price[1]
1	OPCom	ARGO-HYTOS	4, 6, 14, 21	1	max 420 (dyn)	50 - 400	−20 - 85	€
2	OilWear C100	Atten2	4, 6, 14, 21, 38, 70	1	max 150	200 - 500	max 85	€€
3	OilWear S120	Atten2	4, 6, 14, 21, 38, 70	1	max 150	200 - 500	max 85	€€€
4	BPM-100	Bühler	4, 6, 14, 21	1	max 420 (dyn)	50 - 400	−20 - 85	€€
5	CMS 2	Des-Case	4, 6, 14, 21, 25, 38, 50, 70	0.5 - 1	max 420	20 - 400	−25 - 80	-
6	CCM 01	Eaton	4, 6, 14	1	max 50	500 - 50 000	max 70	-
7	PFS 02	Eaton	4, 6, 14	N/A	1.5 - 420	N/A	−20 - 85	-
8	PC9001-3-2	Filtertechnik	4, 6, 14, 21, 38, 50, 70, 100	0.5	max 50	50 - 500	−10 - 60	€
9	S120-PM	Filtertechnik	4, 6, 14, 21, 38, 70	1	max 100	20 - 50	0 - 60	€€
10	OilWear 2.0 (S120)	Filtertechnik	4, 6, 14, 21, 38, 70	1 - 2	max 160	20	0 - 70	€€
11	S120 LCD	Filtertechnik	4, 6, 14, 21, 38, 70	1	max 160	100 - 1000	−20 - 70	€€
12	CS 1939 Series	HYDAC	4, 6, 14, 21	0.5	max 350	30 - 500	0 - 85	€€
13	Patrick	Hydrotechnik	4, 6, 14	1	max 420 (dyn)	50 - 400	−10 - 60	-
14	LDP100	ifm electronic	4, 6, 14, 21	1	max 420 (dyn)	N/A	−10 - 80	€
15	ICS	MP Filtri	4, 6, 14	0.5	max 420	20 - 400	−25 - 100	€[2]
16	ICM 2.0	MP Filtri	4, 6, 14, 21, 25, 38, 50, 70	0.5 - 1	max 420	20 - 400	−25 - 80	€[2]
17	ICU	MP Filtri	4, 6, 14	0.5	25 - 350	200	0 - 80	-
18	OLSP1	Oilense	4, 4.6, 6, 10, 14, 21, 38, 70	1	2 - 100	10 - 300	0 - 80	-
19	PAMAS S50	PAMAS	4, 6, 10, 14, 21, 25, 38, 70	N/A	0.2 - 15	5 - 50	max 60	€€€
20	PAMAS S50P	PAMAS	4, 6, 10, 14, 21, 25, 38, 70	N/A	0 - 6	25	max 60	€€€
21	PAMAS OLS50P	PAMAS	4, 6, 10, 14, 21, 25, 38, 70	N/A	0 - 6	10 or 25	N/A	€€€
22	icountPD	Parker	4, 6, 14	1	2 - 420	40 - 140	5 - 80	€
23	LPM-II-plus	Stauff	4, 6, 14, 21, 25, 38, 50, 70	0.5 - 1	max 420	20 - 400	−25 - 80	-

[a]Price indications are as follows: €= up to 3000 SEK, €€= 3000-6000 SEK, €€€= over 6000 SEK. All prices were converted to SEK and are indicative only.

[b]Based on OEM price.

Table 4.18: Product comparison for in-line particle sensors (continued), showing sensor principal, display availability, supported communication standards, and additional supported fluids.

Pos	Digital Imaging	Display	RS485	RS232	Modbus TC/IP	CAN	4-20mA	Fuels	Water
1	No	Optional	-	Yes	-	Yes	Yes	-	-
2	Yes	No	Yes	-	Yes	-	-	Yes	Yes
3	Yes	Optional	Yes	-	Yes	-	-	Yes	Yes
4	No	Yes	-	Yes	-	Yes	Yes	-	-
5	No	Yes	Yes	-	-	Yes	Yes	-	-
6	No	Yes	-	Yes	-	Yes	-	-	-
7	No	No	-	Yes	-	Yes	Yes	-	-
8	No	Yes	Yes	-	-	-	-	-	-
9	Yes	Yes	-	-	-	-	-	Yes	-
10	Yes	No	Yes	-	Yes	-	-	-	-
11	Yes	Yes	Yes	-	Yes	-	-	Yes	Yes
12	No	No	-	-	-	Yes	-	-	-
13	No	Yes	-	Yes	-	Yes	Yes	Yes	-
14	No	Yes	-	-	-	Yes	-	-	-
15	No	No	Yes	-	-	-	Yes	-	-
16	No	Optional	Yes	-	-	Yes	Yes	-	Optional
17	No	No	-	-	-	-	Yes	-	-
18	No	Yes	Yes	-	-	-	-	-	-
19	No	Yes	Yes	-	-	-	Optional	-	-
20	No	Yes	Yes	-	-	-	Optional	-	-
21	No	No	Yes	-	-	-	Yes	-	Yes
22	No	Yes	-	Yes	-	Yes	Yes	Optional	-
23	No	Yes	Yes	Yes	-	Yes	Yes	-	Yes

5

Conclusion

This research aimed to develop and evaluate predictive maintenance (PdM) models for estimating the Remaining Useful Life (RUL) of oil filters within industrial hydraulic systems, addressing challenges posed by the availability of limited and heterogeneous operational data. A pipeline was developed encompassing data acquisition, preprocessing, feature engineering, model selection, and evaluation strategies tailored for time-series data and limited sample sizes. With this pipeline, various machine learning and deep learning approaches across a proprietary dataset and two public benchmarks were explored.

A critical finding from the analysis of the Primary Oil Filter Dataset was the significant difficulty in achieving reliable and consistent RUL predictions across all operational cycles due to the inherent limitations of the data, particularly the small number of filter cycles (four) and the sparsity of key laboratory measurements (particle counts, MPC values, water content). This dataset highlighted that successful RUL prediction in real-world industrial scenarios is highly dependent on the quality and quantity of available data and the presence of reliable degradation signals.

Specifically, the study revealed two necessary conditions for achieving acceptable predictive performance (defined as consistently positive R^2 scores across all evaluation folds) on the Primary Oil Filter Dataset. First, the target variable definition significantly affected model performance; predicting the Volume-based RUL (remaining cumulative oil volume) yielded substantially more consistent and positive results compared to predicting the Time-based RUL (remaining operational time). The volume of cleaned oil appeared to be a more robust indicator of filter degradation in this specific application. The second necessary condition was the presence of the cumulative oil volume value as a feature in the data. Its inclusion was found to be essential across all investigated feature sets. Without this single feature explicitly available to the models, predictions consistently failed to achieve positive R^2 scores across all folds, underscoring its overwhelming dominance as a predictor in this data-constrained scenario.

While extensive automated feature extraction methods and dimensionality reduction techniques were explored, they did not consistently improved predictive performance on the limited primary dataset. In contrast to their sometimes beneficial effects observed on the more comprehensive complementary datasets, this finding highlights that for very small datasets, careful manual feature selection based on domain knowledge can be more effective than large-scale automated generation. Feature importance based on the tree-based models allowed for the identification of

key contributors to RUL prediction on the primary dataset. Based on this analysis and domain understanding, a curated subset of features was selected, and a final evaluation round was successfully conducted on this reduced feature set, demonstrating improved and more consistent performance for some models.

Conversely, validation efforts using the more comprehensive C-MAPSS and Dust Filter datasets demonstrated the potential of the selected modeling approaches when applied to datasets with a larger number of cycles and clearer degradation patterns. Deep learning architectures, particularly the 1D-CNN and CNN-GRU models, showed strong predictive capabilities on these datasets, achieving high R^2 scores which were comparable to benchmark results from academic literature. This validates the general applicability of the developed pipeline and chosen models under different data conditions.

Furthermore, the market study on inline particle sensors identified several commercially available technologies capable of providing high-frequency particle contamination data. This was identified as a critical data gap in the current Primary Oil Filter Dataset. It was found that the sensor's operating pressure is generally compatible, where as the flow rate limitations of most sensors necessitate the integration of the particle sensor via a filter bypass loop.

In summary, this research highlighted the significant challenges of implementing data-driven PdM with limited, heterogeneous data, pointing out the critical features necessary for prediction in the specific industrial case study, utilized feature importance analysis to guide feature selection for improved performance, validated the potential of the chosen modeling techniques on benchmark datasets, and proposed a concrete technical solution (inline particle sensor with bypass) to acquire the data necessary for more robust predictions in the future. The findings underscore the importance of sufficient, relevant data as the foundation for effective predictive maintenance implementations.

Future Work

Based on the findings and limitations of this research, the following recommendations are made to advance the goal of reliable estimations of RUL for oil filters in industrial fluid systems.

The most critical step for improving predictive accuracy on the Primary Oil Filter Dataset is to enhance the data acquisition system. Based on the market study findings, the potential installation of a selected inline particle sensor is recommended. Given the operating condition constraints, integration could be achieved via a filter bypass loop. This would enable the acquisition of high-frequency data, potentially replacing the infrequent and sparse laboratory particle count measurements. Such a change would dramatically increase the frequency of particle count data, and simultaneously reduce data handling effort associated with offline sampling and analysis. Concurrently, efforts should be made to systematically collect data over a significantly larger number of filter operational cycles, potentially including new machines, to build a more robust and diverse dataset. Working towards establishing more objective, data-driven criteria for filter replacement, rather than relying solely

on expert judgment, would also provide a clearer and more consistent target variable for RUL prediction.

Once a richer dataset becomes available, further model development and refinement can be conducted. The performance of the evaluated machine learning and deep learning models should be re-assessed on the new dataset. While a limited hyperparameter tuning was performed in this study, a more comprehensive and systematic optimization is recommended, including additional hyperparameters. A more detailed analysis of feature importance, potentially utilizing methods such as SHapley Additive exPlanations (SHAP), can be conducted on the new dataset to gain deeper insights into the contribution of individual features to RUL predictions. However, drawing reliable conclusions from such sophisticated analytical techniques depends fundamentally on a sufficiently comprehensive and robust dataset.

Bibliography

[1] Michael M. Khonsari and E. Richard Booser. *Applied Tribology: Bearing Design and Lubrication*. en. 1st ed. Wiley, Apr. 2008. DOI: 10.1002/9780470059456.

[2] Miroslav Mindas, Lucia Knapčíková, and Michal Balog. "Productivity Fluid Management as a Tool for Saving Money in Manufacturing". In: *TEM JOURNAL-TECHNOLOGY EDUCATION MANAGEMENT INFORMATICS* 5 (June 2016). DOI: 10.18421/TEM52-11.

[3] *Cutting Fluid Management: Small Machining Operations*. en. Iowa Waste Reduction Center, 2003.

[4] Andrew K.S. Jardine, Daming Lin, and Dragan Banjevic. "A review on machinery diagnostics and prognostics implementing condition-based maintenance". en. In: *Mechanical Systems and Signal Processing* 20.7 (Oct. 2006), pp. 1483–1510. ISSN: 08883270. DOI: 10.1016/j.ymssp.2005.09.012.

[5] Dian Jiao et al. "Oil property sensing array based on a general regression neural network". en. In: *Tribology International* 164 (Dec. 2021), p. 107221. ISSN: 0301679X. DOI: 10.1016/j.triboint.2021.107221.

[6] Ping Lu et al. "Early wear detection and its significance for condition monitoring". en. In: *Tribology International* 159 (July 2021), p. 106946. ISSN: 0301679X. DOI: 10.1016/j.triboint.2021.106946.

[7] Haichuan Chang, Pietro Borghesani, and Zhongxiao Peng. "Investigation on the relationship between macropits and wear particles in a gear fatigue process". en. In: *Wear* 484-485 (Nov. 2021), p. 203724. ISSN: 00431648. DOI: 10.1016/j.wear.2021.203724.

[8] Ailin Deng and Bryan Hooi. "Graph Neural Network-Based Anomaly Detection in Multivariate Time Series". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 35.5 (May 2021), pp. 4027–4035. ISSN: 2374-3468, 2159-5399. DOI: 10.1609/aaai.v35i5.16523.

[9] Owais Asif et al. "A Deep Learning Model for Remaining Useful Life Prediction of Aircraft Turbofan Engine on C-MAPSS Dataset". en. In: *IEEE Access* 10 (2022), pp. 95425–95440. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2022.3203406.

[10] Hui Tao et al. "A Wear Condition Warning Method for Wind Turbine Gearbox Based on Oil Online Monitoring Using Learnable Multiscale Convolutional Neural Network". In: *IEEE Sensors Journal* 24.21 (Nov. 2024). Conference Name: IEEE Sensors Journal, pp. 35709–35721. ISSN: 1558-1748. DOI: 10.1109/JSEN.2024.3462815.

[11] Zakwan Skaf, Omer F. Eker, and Ian K. Jennions. "A Simple State-Based Prognostic Model for Filter Clogging". In: *Procedia CIRP*. Proceedings of the 4th International Conference on Through-life Engineering Services 38 (Jan. 2015), pp. 177–182. ISSN: 2212-8271. DOI: 10.1016/j.procir.2015.08.094.

[12] Seunghyun Lee et al. "Data-driven health condition and RUL prognosis for liquid filtration systems". en. In: *Journal of Mechanical Science and Technology* 35.4 (Apr. 2021), pp. 1597–1607. ISSN: 1738-494X, 1976-3824. DOI: 10.1007/s12206-021-0323-8.

[13] Nikolai Helwig, Eliseo Pignanelli, and Andreas Schütze. "Condition monitoring of a complex hydraulic system using multivariate statistics". In: *2015 IEEE International Instrumentation and Measurement Technology Conference (I2MTC) Proceedings*. ISSN: 1091-5281. May 2015, pp. 210–215. DOI: 10.1109/I2MTC.2015.7151267.

[14] Wenbo Han et al. "A Critical Review of On-Line Oil Wear Debris Particle Detection Sensors". en. In: *Journal of Marine Science and Engineering* 11.12 (Dec. 2023), p. 2363. ISSN: 2077-1312. DOI: 10.3390/jmse11122363.

[15] International Organization for Standardization. *ISO 4406:2021 – Hydraulic fluid power – Fluids – Method for coding the level of contamination by solid particles*. International Standard. Geneva, Switzerland: International Organization for Standardization, 2021.

[16] Kevin Krogsøe, Morten Henneberg, and René Lynge Eriksen. "Model of a Light Extinction Sensor for Assessing Wear Particle Distribution in a Lubricated Oil System". en. In: *Sensors* 18.12 (Nov. 2018), p. 4091. ISSN: 1424-8220. DOI: 10.3390/s18124091.

[17] D02 Committee. *Test Method for Automatic Particle Counting and Particle Shape Classification of Oils Using a Direct Imaging Integrated Tester*. en. DOI: 10.1520/D7596-23.

[18] Jon Mabe, Joseba Zubia, and Eneko Gorritxategi. "Photonic Low Cost Micro-Sensor for in-Line Wear Particle Detection in Flowing Lube Oils". en. In: *Sensors* 17.3 (Mar. 2017), p. 586. ISSN: 1424-8220. DOI: 10.3390/s17030586.

[19] Farzaneh Ahmadzadeh and Jan Lundberg. "Remaining useful life estimation: review". en. In: *International Journal of System Assurance Engineering and Management* 5.4 (Dec. 2014), pp. 461–474. ISSN: 0975-6809, 0976-4348. DOI: 10.1007/s13198-013-0195-0.

[20] Mahmoud Rahat et al. "Bridging the Gap: A Comparative Analysis of Regressive Remaining Useful Life Prediction and Survival Analysis Methods for Predictive Maintenance". en. In: *PHM Society Asia-Pacific Conference* 4.1 (Sept. 2023). ISSN: 2994-7219, 2994-7219. DOI: 10.36001/phmap.2023.v4i1.3646.

[21] Jonathon Shlens. *A Tutorial on Principal Component Analysis*. arXiv:1404.1100 [cs]. Apr. 2014. DOI: 10.48550/arXiv.1404.1100.

[22] Richard A. Berk. "An Introduction to Ensemble Methods for Data Analysis". en. In: *Sociological Methods & Research* 34.3 (Feb. 2006), pp. 263–295. ISSN: 0049-1241, 1552-8294. DOI: 10.1177/0049124105283119.

[23] Leo Breiman. "Random Forests". In: *Machine Learning* 45.1 (2001), pp. 5–32. ISSN: 08856125. DOI: 10.1023/A:1010933404324.

[24] Jerome H. Friedman. "Greedy function approximation: A gradient boosting machine." In: *The Annals of Statistics* 29.5 (Oct. 2001). ISSN: 0090-5364. DOI: 10.1214/aos/1013203451.

[25] Yoav Freund and Robert E Schapire. "A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting". en. In: *Journal of Computer and System Sciences* 55.1 (Aug. 1997), pp. 119–139. ISSN: 00220000. DOI: 10.1006/jcss.1997.1504.

[26] Osval Antonio Montesinos López, Abelardo Montesinos López, and José Crossa. *Multivariate Statistical Machine Learning Methods for Genomic Prediction*. en. Cham: Springer International Publishing, 2022. ISBN: 978-3-030-89009-4 978-3-030-89010-0. DOI: 10.1007/978-3-030-89010-0.

[27] Shubhi Harbola and Volker Coors. "One dimensional convolutional neural network architectures for wind prediction". en. In: *Energy Conversion and Management* 195 (Sept. 2019), pp. 70–75. ISSN: 01968904. DOI: 10.1016/j.enconman.2019.05.007.

[28] Jeff Heaton. "Ian Goodfellow, Yoshua Bengio, and Aaron Courville: Deep learning: The MIT Press, 2016, 800 pp, ISBN: 0262035618". en. In: *Genetic Programming and Evolvable Machines* 19.1-2 (June 2018), pp. 305–307. ISSN: 1389-2576, 1573-7632. DOI: 10.1007/s10710-017-9314-z.

[29] Kyunghyun Cho et al. *Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation*. Version Number: 3. 2014. DOI: 10.48550/ARXIV.1406.1078.

[30] Shuhan Sun et al. "Few-shot RUL prediction for engines based on CNN-GRU model". en. In: *Scientific Reports* 14.1 (July 2024), p. 16041. ISSN: 2045-2322. DOI: 10.1038/s41598-024-66377-3.

[31] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[32] David S. Moore, William Notz, and Michael A. Fligner. *The basic practice of statistics*. eng. Eighth edition. OCLC: 1002669890. New York: W.H. Freeman, 2018. ISBN: 978-1-319-04257-8.

[33] Abhinav Saxena Saxena. *NASA Turbofan Jet Engine Data Set*. DOI: 10.21227/PJH5-P424.

[34] Emmanuel Ramasso and Abhinav Saxena. "Performance Benchmarking and Analysis of Prognostic Methods for CMAPSS Datasets". In: *International Journal of Prognostics and Health Management* 5 (Nov. 2014), pp. 1–15. DOI: 10.36001/ijphm.2014.v5i2.2236.

[35] Simon Vollert and Andreas Theissler. "Challenges of machine learning-based RUL prognosis: A review on NASA's C-MAPSS data set". In: *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. Sept. 2021, pp. 1–8. DOI: 10.1109/ETFA45728.2021.9613682.

[36] Simon Hagmeyer, Fabian Mauthe, and Peter Zeiler. "Creation of Publicly Available Data Sets for Prognostics and Diagnostics Addressing Data Scenarios Relevant to Industrial Applications". en. In: *International Journal of Prognostics and Health Management* 12.2 (Nov. 2021). Number: 2. ISSN: 2153-2648. DOI: 10.36001/ijphm.2021.v12i2.3087.

[37] A. M. Stolwijk, H. Straatman, and G. A. Zielhuis. "Studying seasonality by using sine and cosine functions in regression analysis". en. In: *Journal of Epidemiology & Community Health* 53.4 (Apr. 1999), pp. 235–238. ISSN: 0143-005X. DOI: 10.1136/jech.53.4.235.

[38] Aayam Bansal, Keertan Balaji, and Zeus Lalani. *Temporal Encoding Strategies for Energy Time Series Prediction*. Version Number: 1. 2025. DOI: 10.48550/ARXIV.2503.15456.

[39] Maximilian Christ et al. "Time Series FeatuRe Extraction on basis of Scalable Hypothesis tests (tsfresh – A Python package)". en. In: *Neurocomputing* 307 (Sept. 2018), pp. 72–77. ISSN: 09252312. DOI: 10.1016/j.neucom.2018.03.067.

[40] *tsfresh/tsfresh/feature_extraction/settings.py at main · blue-yonder/tsfresh*. en. URL: https://github.com/blue-yonder/tsfresh/blob/main/tsfresh/feature_extraction/settings.py.

[41] M. Grasso, B.M. Colosimo, and M. Pacella. "Profile monitoring via sensor fusion: the use of PCA methods for multi-channel data". en. In: *International Journal of Production Research* 52.20 (Oct. 2014), pp. 6110–6135. ISSN: 0020-7543, 1366-588X. DOI: 10.1080/00207543.2014.916431.

[42] Maz Jamilah Masnan et al. "Principal component analysis–a realization of classification success in multi sensor data fusion". In: *Principal Component Analysis Application* (2012), pp. 1–25.

[43] Yazan Alomari, Mátyás Andó, and Marcia L. Baptista. "Advancing aircraft engine RUL predictions: an interpretable integrated approach of feature engineering and aggregated feature importance". en. In: *Scientific Reports* 13.1 (Aug. 2023), p. 13466. ISSN: 2045-2322. DOI: 10.1038/s41598-023-40315-1.

[44] François Chollet. *Keras*. 2015. URL: https://keras.io.

[45] Martín Abadi et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. URL: https://www.tensorflow.org/.

[46] Kıymet Ensarioğlu, Tülin İnkaya, and Erdal Emel. "Remaining Useful Life Estimation of Turbofan Engines with Deep Learning Using Change-Point Detection Based Labeling and Feature Engineering". en. In: *Applied Sciences* 13.21 (Oct. 2023), p. 11893. ISSN: 2076-3417. DOI: 10.3390/app132111893.

[47] Daniele Pochi et al. "Bench Testing of Sensors Utilized for In-Line Monitoring of Lubricants and Hydraulic Fluids Properties". en. In: *Sensors* 21.24 (Dec. 2021), p. 8201. ISSN: 1424-8220. DOI: 10.3390/s21248201.

[48] Vignesh V. Shanbhag et al. "Failure Monitoring and Predictive Maintenance of Hydraulic Cylinder—State-of-the-Art Review". In: *IEEE/ASME Transactions on Mechatronics* 26.6 (Dec. 2021), pp. 3087–3103. ISSN: 1941-014X. DOI: 10.1109/TMECH.2021.3053173.

[49] Haotian Shi et al. "Capacitive–Inductive Magnetic Plug Sensor With High Adaptability for Online Debris Monitoring". In: *IEEE Transactions on Instrumentation and Measurement* 71 (2022). Conference Name: IEEE Transactions on Instrumentation and Measurement, pp. 1–8. ISSN: 1557-9662. DOI: 10.1109/TIM.2022.3156981.

[50] Oluseun Omotola Aremu et al. "A relative entropy based feature selection framework for asset data in predictive maintenance". In: *Computers & Indus-*

trial Engineering 145 (July 2020), p. 106536. ISSN: 0360-8352. DOI: 10.1016/j.cie.2020.106536.

[51] Hanchuan Peng, Fuhui Long, and C. Ding. "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 27.8 (Aug. 2005), pp. 1226–1238. ISSN: 1939-3539. DOI: 10.1109/TPAMI.2005.159.

A

Appendix 1

A.1 Key Statistics of the Primary Oil Filter Dataset

Table A.1: Key Statistics for Sensor Data of the Primary Oil Filter Dataset.

Variable	Filter Cycle 1	Filter Cycle 2	Filter Cycle 3	Filter Cycle 4
Flow ($\mathrm{L\,min^{-1}}$)	0.35 0.75 1.00	0.87 1.11 1.41	0.56 0.89 1.18	2.47 3.70 4.11
Pressure (bar)	2.40 2.50 2.63	0.90 0.99 1.09	0.93 1.00 1.08	2.43 2.50 2.59
Temp (°C)	22.50 36.11 41.70	17.30 21.22 26.30	15.60 21.07 22.80	17.30 19.64 21.20
RH (%)	0.00 1.03 3.25	2.51 2.81 3.08	2.50 2.84 3.25	2.61 2.90 3.17

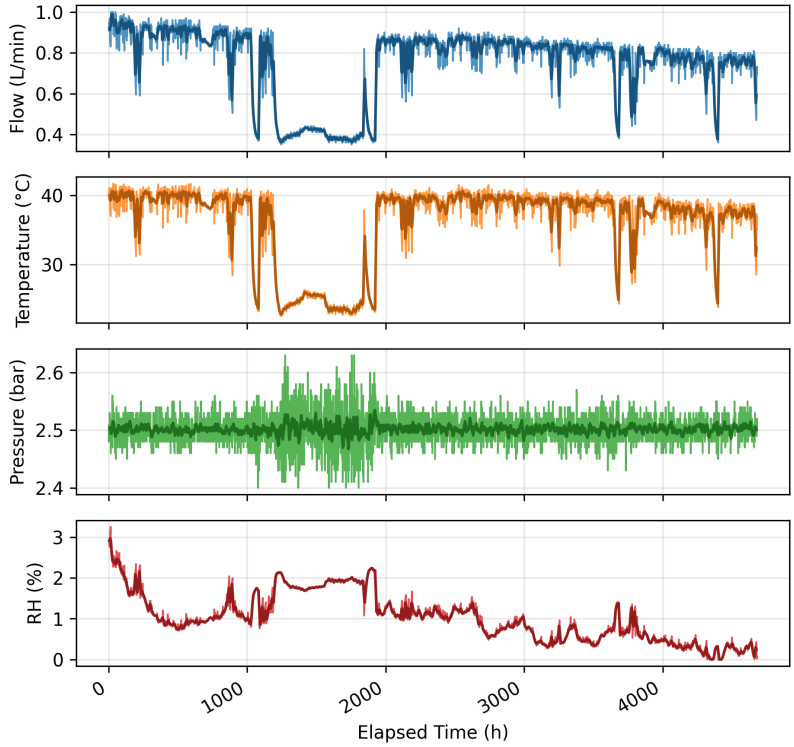
Statistics: Top = min, Middle = mean, Bottom = max

Table A.2: Key Statistics for Laboratory Data of the Primary Oil Filter Dataset.

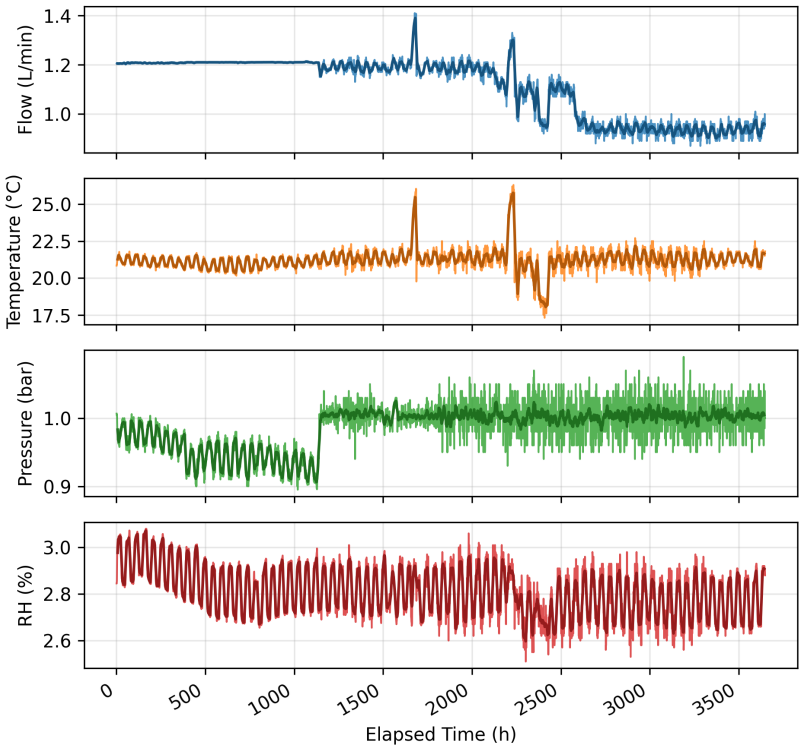
Variable	Filter Cycle 1	Filter Cycle 2	Filter Cycle 3	Filter Cycle 4
	76.00	9948.12	1616.00	312.00
4um	147.56	42 256.90	41 992.28	9164.59
	757.26	76 548.33	87 751.33	27 072.06
	9.33	1740.31	331.67	160.00
6um	34.68	11 705.97	14 587.72	2805.04
	243.76	29 350.97	34 981.67	7946.32
	2.00	160.00	160.00	160.00
14um	4.39	1043.99	1116.82	170.34
	20.41	4452.39	5173.33	223.86
	4.94	3.00	4.00	0.00
MPC	7.53	8.34	8.91	2.95
	34.16	16.51	12.99	6.63
	3.00	23.00	5.00	12.07
Water (ppm)	20.56	29.31	20.99	21.99
	34.98	45.69	28.00	30.00

Statistics: Top = min, Middle = mean, Bottom = max

A.2 Historical Plots of the Primary Oil Filter Dataset

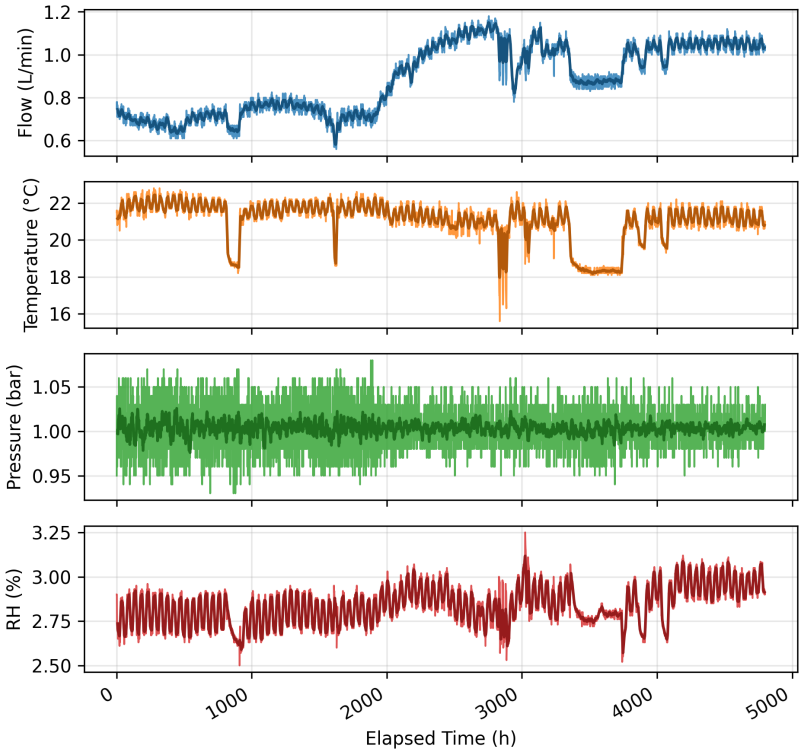


(a) Filter Cycle 1

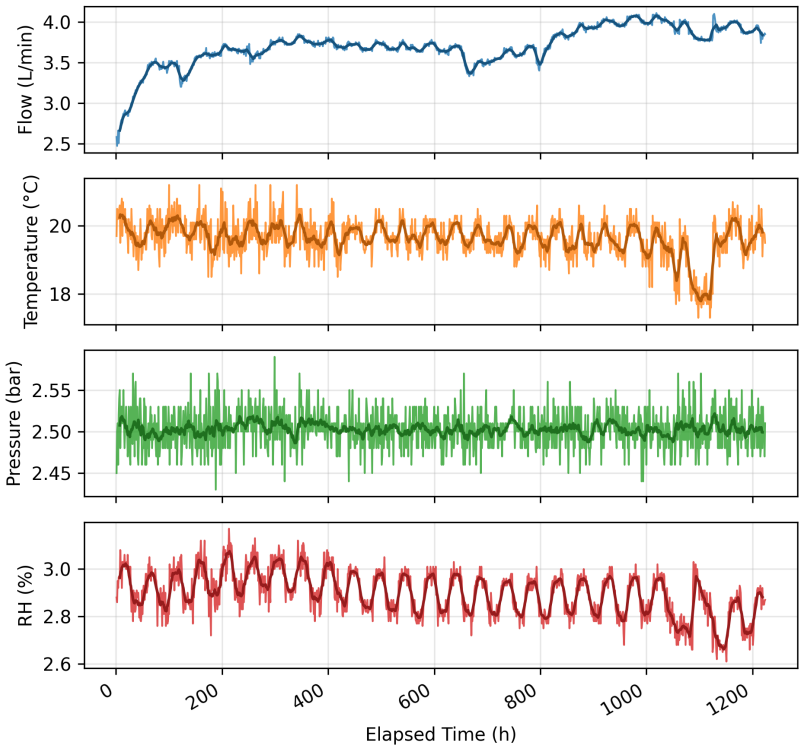


(b) Filter Cycle 2

Figure A.1: Sensor data over the cycle time for all four cycles of the Primary Oil Filter Dataset (enlarged, part 1/2).

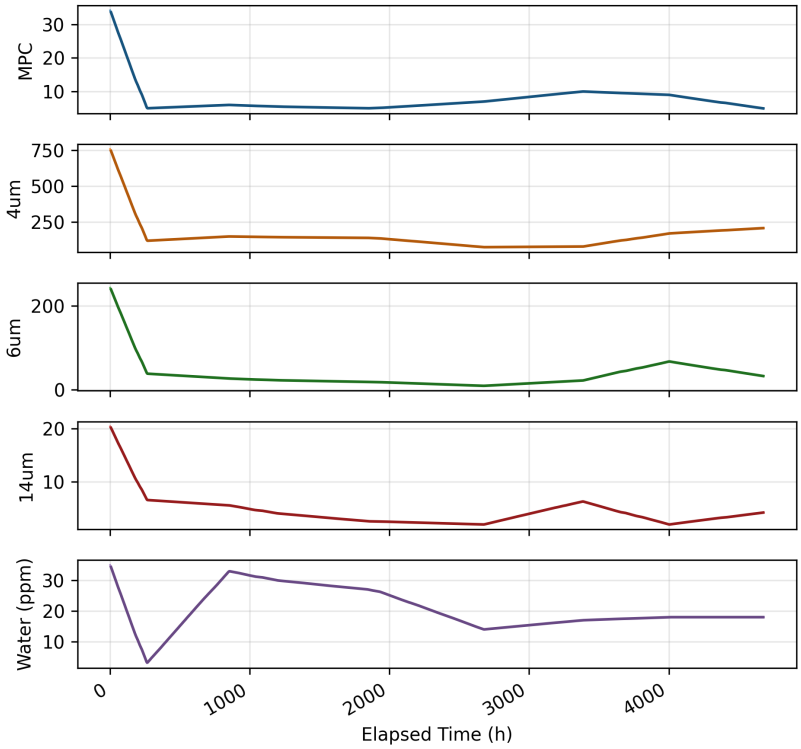


(c) Filter Cycle 3

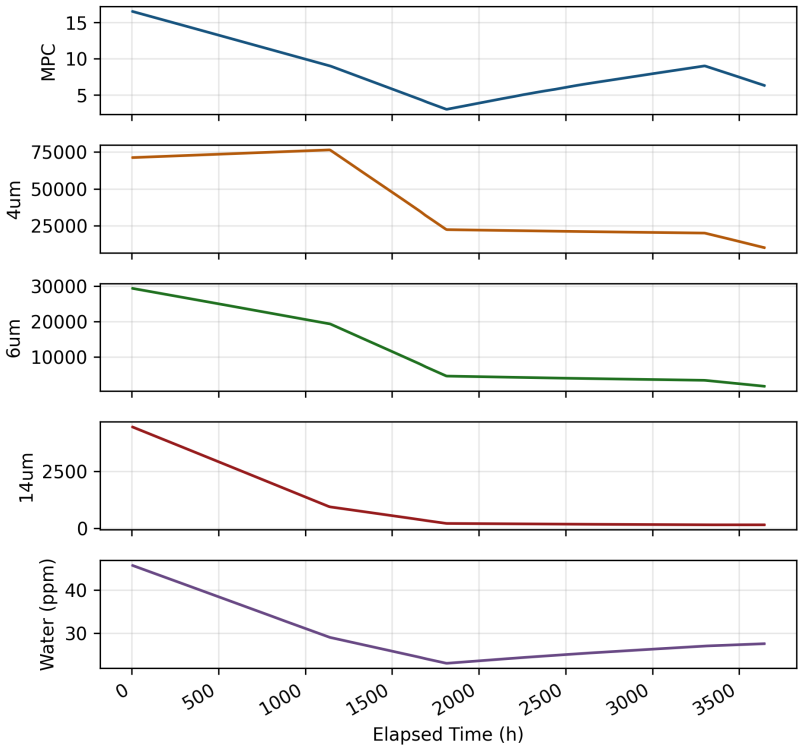


(d) Filter Cycle 4

Figure A.1: Sensor data over the cycle time for all four cycles of the Primary Oil Filter Dataset (enlarged, part 2/2).

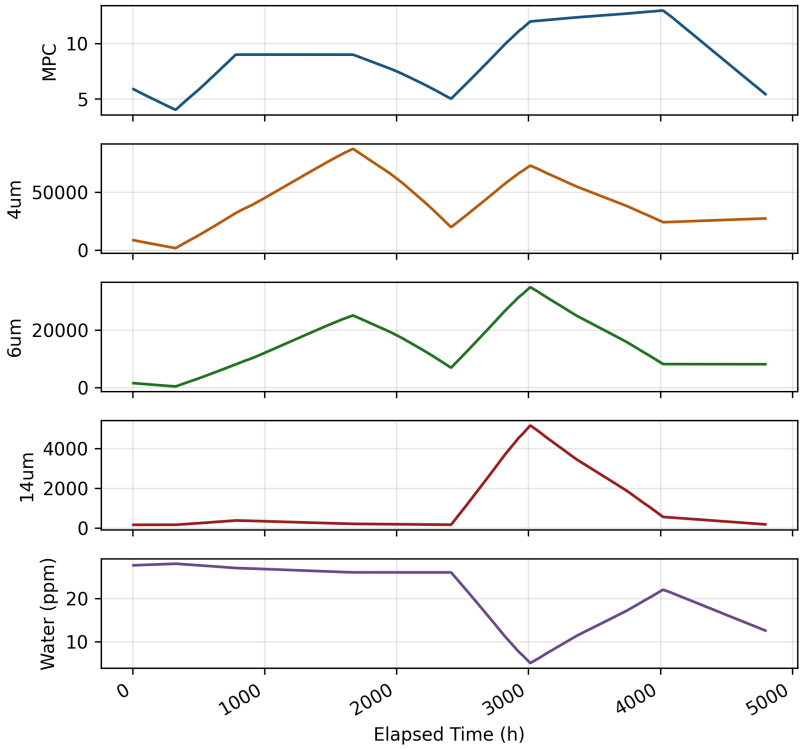


(a) Filter Cycle 1

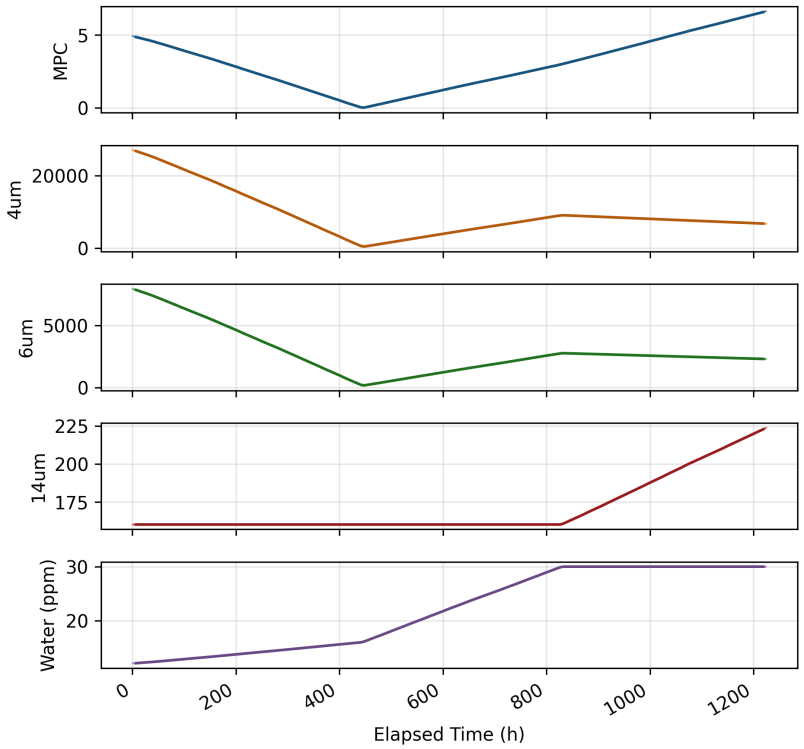


(b) Filter Cycle 2

Figure A.2: Laboratory measurement data over the cycle time for all four cycles of the Primary Oil Filter Dataset (enlarged, part 1/2).



(c) Filter Cycle 3



(d) Filter Cycle 4

Figure A.2: Laboratory measurement data over the cycle time for all four cycles of the Primary Oil Filter Dataset (enlarged, part 2/2).

A.3 Correlation Matrices

Correlation Matrices Primary Oil Filter Dataset

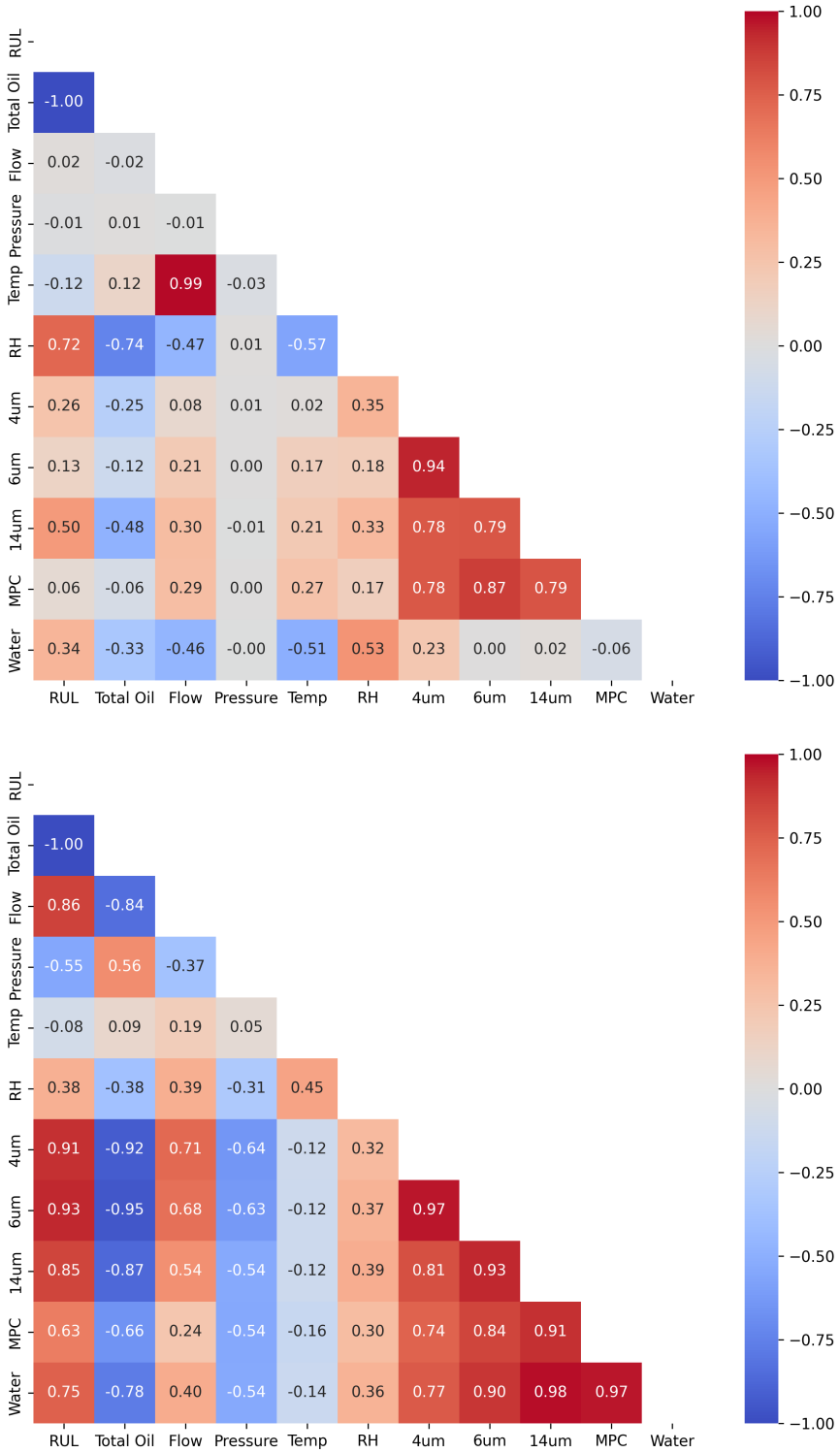


Figure A.3: Individual sensor correlation for all four cycles of Primary Oil Filter Dataset (enlarged, part 1/2).

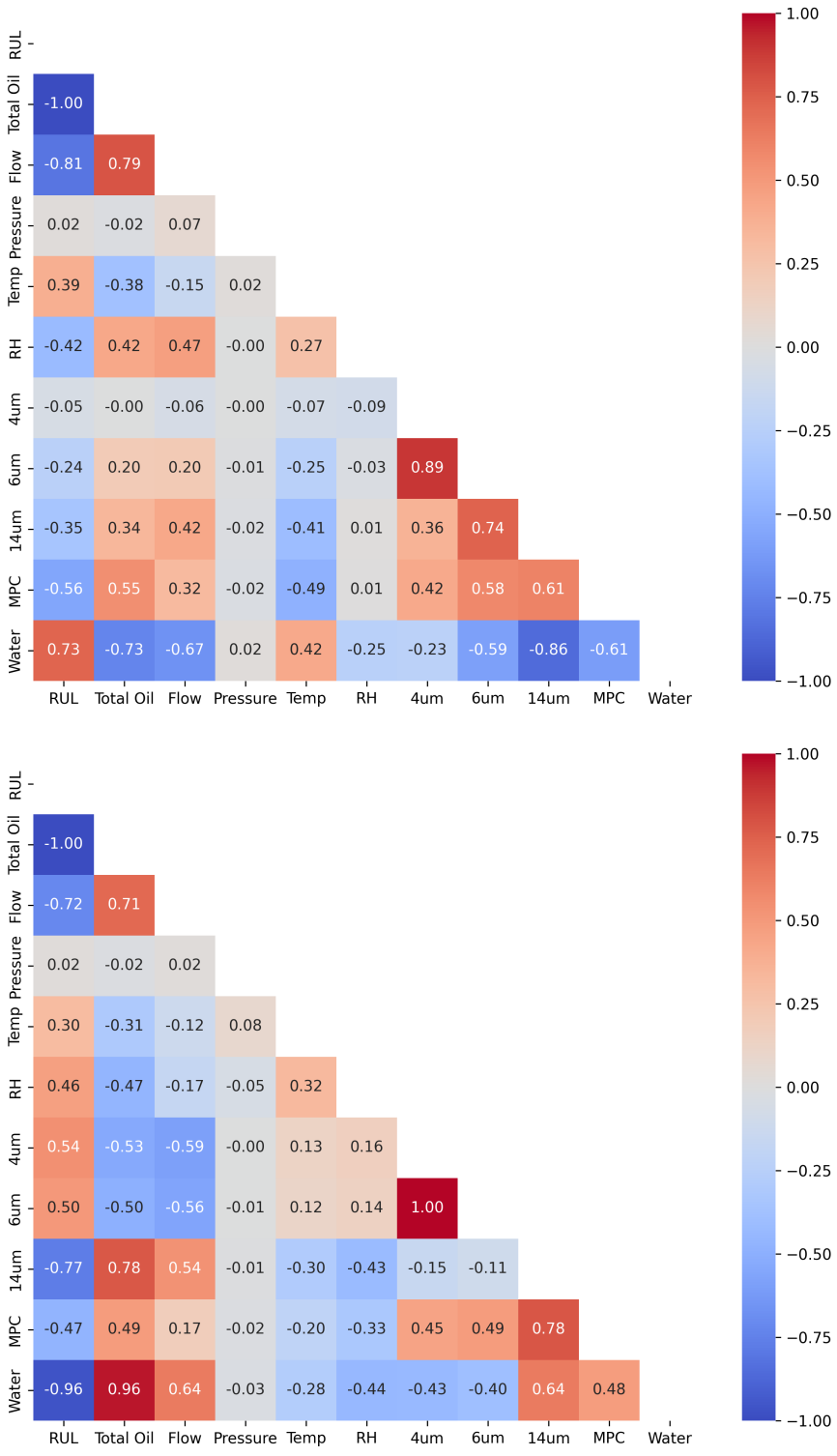


Figure A.3: Individual sensor correlation for all four cycles of Primary Oil Filter Dataset (enlarged, part 2/2).

Correlation Matrices C-MAPSS Dataset

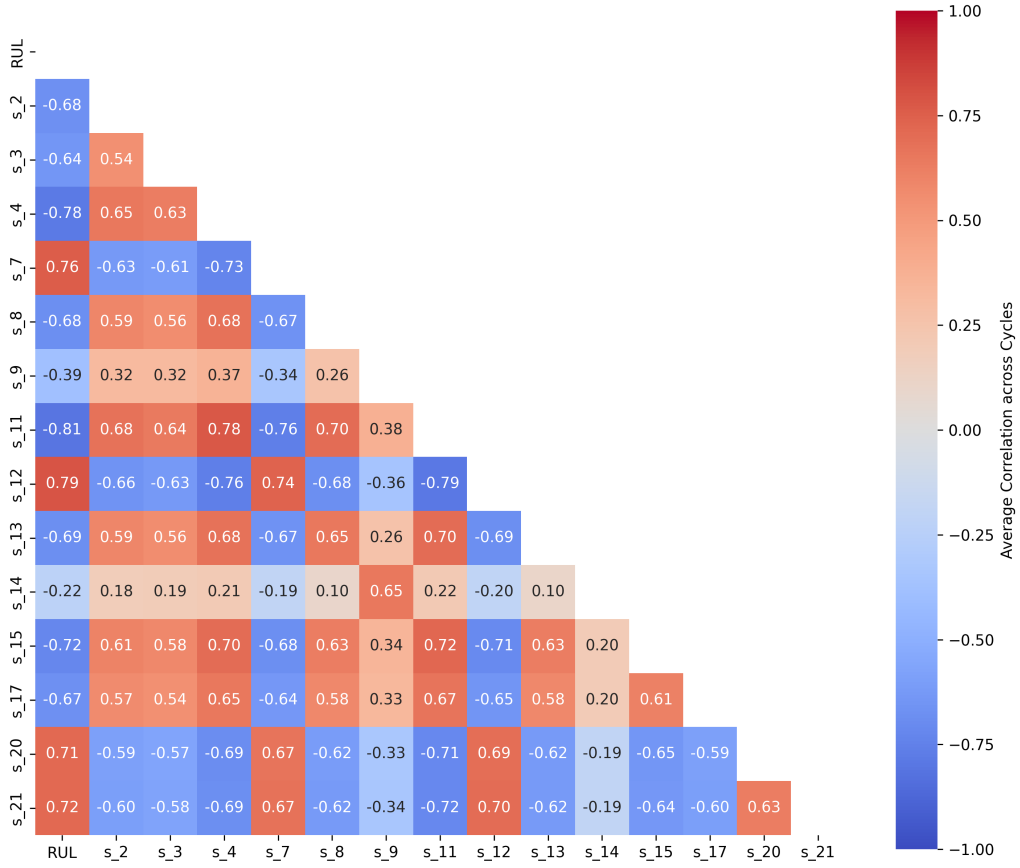


Figure A.4: Average Pearson correlation coefficients between sensors and RUL for the C-MAPSS Dataset.

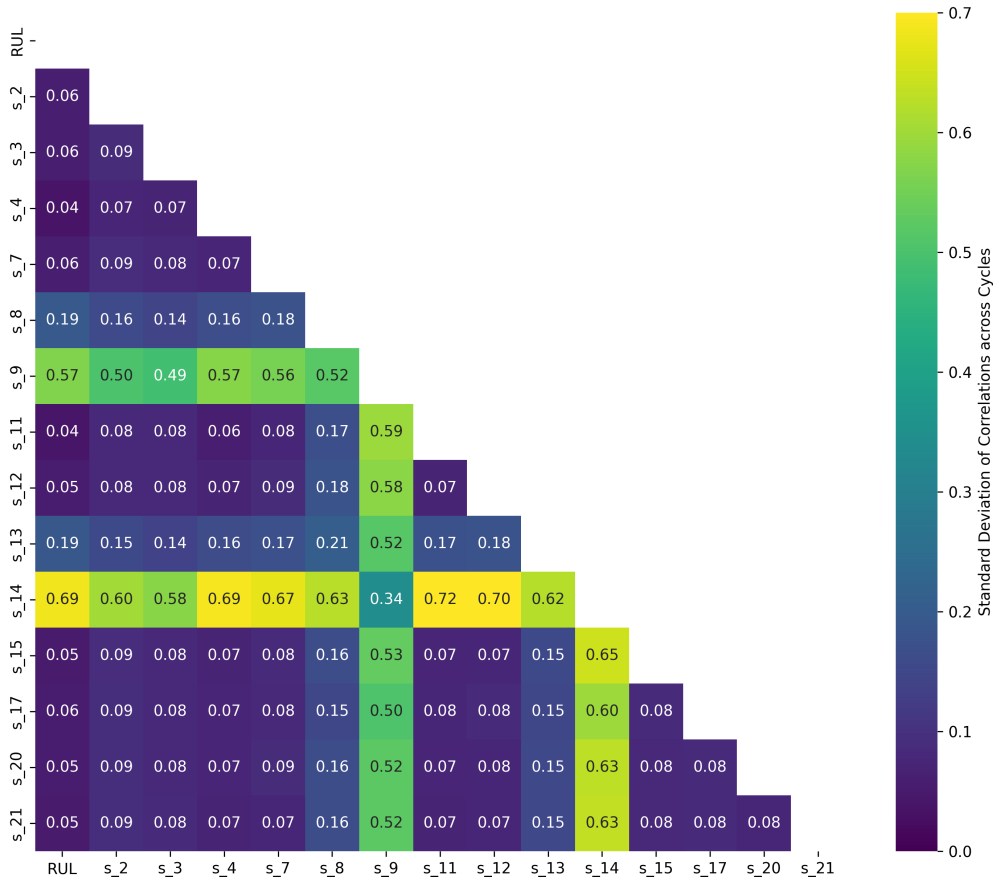


Figure A.5: Standard deviation of Pearson correlation coefficients between sensors and RUL for the C-MAPSS Dataset.

DEPARTMENT OF INDUSTRIAL AND MATERIALS SCIENCE
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY