Prognostic Distributions for the SOH and RUL of EVTOLs

Leo Jenneskens

March 2024

Abstract

In this thesis, we have explored the utilization of different algorithms for prognostic distribution for the state of the health of batteries of eVTOLS. During the research we will be exploring different algorithms, such as Convolutional Neural Network(CNN), Long Short Term Memory neural network(LSTM), Bayesian Neural Network(BNN) and Mixture Density Network(MDN) implementation for the prediction of the prognostic distributions for different representation of the health of the batteries in terms of State of Health(SOH) and Remaining Useful life(RUL). Additionally, this study will delve into a novel interpretation of the RUL concept tailored specifically to eVTOLS. The use of these algorithms could have the potential of unlocking new ways for prediction of prognostic of the batteries of eVTOLS. Including more in depth evaluation of these algorithms compared to more generic metrics. In this thesis, we present the results of these algorithms for the prognostic prediction of these distributions utilizing multiple approaches.

Contents

1 Introduction **5**
- 1.1 Problem description . . . 5
- 1.2 Thesis Summary . . . 6

2 Literature Review **6**
- 2.1 Convolutional Neural Network . . . 6
- 2.2 Bayesian Neural Network . . . 7
- 2.3 Multi-Layer Perceptron . . . 7
- 2.4 Long Short Term Memory . . . 8
- 2.5 Mixture Density Network . . . 8

3 Data Description - batteries for eVTOLS **9**
- 3.1 Mission Profile . . . 9
  - 3.1.1 Contrasting Mission Profiles . . . 10
- 3.2 Data Specifications . . . 10
- 3.3 Impedance Data . . . 11
- 3.4 Capacity Tests . . . 12
  - 3.4.1 Definitions of SOH, RUL and RUC . . . 13
- 3.5 Mission Selection . . . 13
  - 3.5.1 VAH07 . . . 15
  - 3.5.2 Missing data . . . 15

4 Methodology for data-driven prognostics of SOH, RUL and RUC **15**
- 4.1 Features . . . 15
- 4.2 Framework for prognostics of SOH, RUL and RUC using machine learning . . . 17
  - 4.2.1 Feature importance . . . 18
  - 4.2.2 Predicting the distribution of SOH, RUL, RUC - probabilistic prognostics . . . 18
  - 4.2.3 Leave one out - Cross Validation . . . 19
  - 4.2.4 Preproccesing . . . 20
  - 4.2.5 Hyperparameter tuning . . . 20
  - 4.2.6 Evaluation Metrics . . . 22

5 Overall results **24**

6 Estimating the distribution of SOH (probabilistic prognostics) **25**
- 6.1 Hypertuning results SOH . . . 25
- 6.2 Results SOH . . . 27

7 Estimating the distribution of RUL (probabilistic prognostics) **31**
- 7.1 Hypertuning results RUL . . . 31
- 7.2 Results RUL . . . 33

8 Estimating the distribution of the RUC (probabilistic prognostics) 36
8.1 Hypertuning results RUC 36
8.2 results RUC 38
8.3 RUC as Conversion of the RUL 40

9 Conclusion 41

10 Appendix A 43
10.1 Appendix A-1: Overview of mission profile specifications 43
10.2 Appendix A-2: feature importance based on the shap values 44
10.3 Appendix A-3: SOH results generic metrics 45
10.4 Appendix A-4: SOH results Novel metrics 46
10.5 Appendix A-5: RUL results generic metrics 48
10.6 Appendix A-6: RUL results Novel metrics 49
10.7 Appendix A-7: RUC results generic metrics 51
10.8 Appendix A-8: RUC results Novel metrics 52

List of Abbreviations

PHM	Prognostics and Health Management
eVTOLS	Electric Vertical Take-off and Landing Aircraft
CNN	Convolutional Neural Network
LSTM	Long Short Term Memory
BNN	Bayesain Neural Network
SOH	State Of Health
SOC	State Of Charge
RUL	Remaining Useful Life
RUC	Remaining Useful Capacity Tests
EOL	End Of Life
MAE	Mean Absolute Error
RMSE	Root Mean Squared Error
CRPS	Continuous Ranked Probability Score
$CRPS_w$	Continuous Ranked Probability Score Weighted
RS over	Reliability Score Overestimation
RS under	Reliability Score Underestimation
stddev	Standard Deviation

Table 1: Table of Abbreviations

1 Introduction

In response to future increase in urban mobility demands and traffic congestion, the adoption of electric short-range aircrafts, such as electric Vertical Take-off and Landing (eVTOLS) aircrafts are increasingly regarded as a promising solution. These eVTOLS are often designed for distances of 50-100km and an average payload of 500-800kg [Pol+19]. The most used battery technology for eVTOLS is Lithium-Ion, due to the the high energy density and low self-discharge as well as feasible costs[Mit+23]. Managing batteries poses a significant challenge for eVTOLS operations. Specifically, ensuring continuous monitoring of the battery's state-of-health is decisive to ensure the safety and efficiency of eVTOL operations. There are different studies for battery management for on-ground Electric vehicles using Lithium-Ion, and the prognostic prediction of the State of Health(SOH) and Remaining Useful Life(RUL). Compared to on-ground e-vehicles, eVTOLS have three flight phases, take-off, cruise and landing . The take-off and landing phases are critical, due to the increased battery discharge compared to the cruise phase. The increased discharge has a direct impact on the SOH of the Lithium-Ion batteries. The aim of this thesis is to compare various methods for computing the regression problem in the field of prognostic health management, specifically the computation of the prognostic distribution for the SOH and RUL of Lithium-Ion batteries used by eVTOLS. The State of Health of a battery reflects its capacity to store and deliver energy compared to its original design capacity in percentage. The RUL is defined as the number of uses(flights) left before the battery needs to be replaced. When the RUL of a component reaches zero, it indicates that the component as reached the end of its lifespan and requires replacement. For the batteries of eVTOLS it has been stated that the RUL equals zero when the SOH reaches 85%. This thesis will utilize data from the Sony-Murata 18650 VTC-6 cell lithium-ion batteries of Vanaha eVTOL from Airbus. Currently, limited research has been conducted on the computation of RUL and SOH prediction distributions, with most studies using the C-MAPPS dataset, which contains data from simulations of turbofans published by NASA. This thesis aims to address this gap in research by focusing on the problem of computing the prognostic distributions for the SOH and RUL of eVTOLS batteries.

1.1 Problem description

To provide a solution for the research gap, we will provide a comparison of multiple methods for the computation of predicted prognostic distributions. In prognostic health management (PHM), it is necessary to prioritize the safety of individual components. For eVTOLS specifically, the safety of the vehicle needs to be ensured. The prognostic distributions of SOH and RUL estimation give a more comprehensive understanding of the health of the Lithium-Ion batteries in the eVTOLS. The goal of this thesis project is to compare current algorithms for the Vahana eVTOLS dataset. At this moment in time, most research is conducted with the C-MAPPS dataset. No research has been undertaken concerning prognostic distributions for eVTOLS. The goal of this thesis can be formulated in the following questions.

How do various approaches for computing prognostic distributions compare with one another for eVTOLS?

To answer the research question the following subquestions will be answered:

- How to implement CNN for the prediction of prognositc distributions for eVTOLS?
- How can we deploy BNN for the prediction of prognostic distributions for eVTOLS?
- How can LSTM be implemented for the prediction of prognostics distributions for EVTOLs?

- How can an MDN implementation be used for the prediction of prognostic distributions?
- How does the prediction of prognostic distribution for the RUL compare to the RUC tests?

1.2 Thesis Summary

In section 2 we will be discussing background studies. In section 3 we will be discussing the Vanaha dataset, the different mission profiles and the description of capacity tests. section 4 we will be discussing retreived features form the Vahana dataset and we will introducing the proposed framework for predicting prognostic distributions and evaluation of these distributions.In section 5 the overall results are discussed for the SOH, RUL and RUC. In section 6 we will be discussing the results for the prognostic predictions for the SOH in more detail, followed by section 7 the results for the prognostic predictions of the RUL in more detail and in section 8 we will be discussing the results for the prognostic distributions of the RUC in more detail.

2 Literature Review

Prognostic distribution prediction has become an important area of research in Prognostics and Health Management (PHM) for RUL prediction. Although numerous studies have been conducted on RUL point prediction, less attention has been given to predicting the distribution of RUL. In this thesis, we aim to address this gap in the literature.

2.1 Convolutional Neural Network

Recent studies have explored the use of Convolutional Neural Networks (CNNs) for RUL point prediction. For instance, Li et al. [LDS18] proposed a CNN model for predicting RUL of turbofans without prior expertise and signal processing. Their study demonstrated that the implementation of a CNN model outperformed other machine learning techniques, such as Neural Networks (NNs), Deep Neural Networks (DNNs), Recurrent Neural Networks (RNNs), and Long-Short Term Memory (LSTM), when evaluated on the C-MAPPS dataset. To mitigate overfitting, Monte Carlo (MC) dropout was used during training. The CNN architecture was designed with five convolutional layers, which strike a balance between training time and Root Mean Squared Error (RMSE) score. Increasing the number of layers further reduced the RMSE score, but this was at the expense of linearly increasing the training time. The results showed that the CNN model performed better than the other machine learning methods while maintaining reasonable training times. In the article [PM22], the authors introduced new metrics to evaluate the accuracy, sharpness, and reliability of prognostic distributions. Unlike general metrics such as Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE), these metrics are specifically designed for the evaluation of distributions. The Continuous Ranked Probability Score (CRPS) was proposed as a metric for evaluating the accuracy and sharpness of the predicted distributions. In addition, two metrics, namely α-Coverage and Reliability Score, were introduced for the evaluation of the reliability of the predicted distributions. To demonstrate the effectiveness of these metrics, the authors applied them to evaluate the distributions estimated by using a CNN model with MC dropout, which was based on the architecture proposed by Li et al. in [LDS18]. The prognostic distributions were created by predicting each test instance multiple times $M_i > 1$ with the neurons randomly dropped during each prediction. The study demonstrated that it is possible to generate prognostic distributions using MC dropout during prediction. This finding generates interest in other methods for using neural networks to generate prognostic distributions. The proposed metrics provide a comprehensive evaluation of the accuracy,

Metric	D3_BP	C2P2_H2
RMSE	13.84	17.76
MAE	9.69	13.22

Table 2: RMSE and MAE of D3_BP(BNN) and C2P2_H2(CNN) for the FD001 subset

sharpness, and reliability of these distributions, which can be used to assess the performance of prognostic models in practical applications.

2.2 Bayesian Neural Network

In the article [Ben+21], a novel approach for predicting distributions using neural networks is proposed. Specifically, the authors introduce Bayesian Neural Networks (BNNs) for predicting distributions in the C-MAPPS dataset. Compared to traditional neural network models, BNNs are different in that they represent the weights and output as probability distributions rather than single values. BNNs create a distribution over the predictions as output, which allows for better estimation of the uncertainty associated with the predictions. In the study, the performance of BNNs is compared to that of Convolutional Neural Networks (CNNs) for predicting the RUL of turbofans in the C-MAPPS dataset. The results show that the proposed BNN approach outperforms the CNN approach in terms of prediction accuracy and uncertainty estimation. Table 2 provides a visual representation of the performance comparison between CNN and BNN in terms of different evaluation metrics. The study highlights the potential of BNNs for predicting distributions in various domains where neural network models are utilized. In [RDH22], the authors use a different approach by implementing a Bayesian Neural Network (BNN) for the C-MAPPS dataset. They utilize a time window-based solution, where the distributions generated by the BNN are used to create a lower bound 95% confidence interval for each engine in the dataset. Unlike [Ben+21], where the predicted RUL is determined by the prediction with the highest probability, in [RDH22], the authors use the distribution to decide whether maintenance is required based on whether the BNN is over or under predicting the RUL. The authors compare their method with other state-of-the-art methods and demonstrate that the BNN can achieve comparable results. This approach has the advantage of providing a more nuanced view of the predicted RUL by offering a confidence interval, which can help in making informed decisions regarding maintenance scheduling.

2.3 Multi-Layer Perceptron

In the article [Kim21], the authors introduce several novel applications for Multi-Layer Perceptron (MLP) models with MC dropout. Dropout is used as a regularization technique to prevent overfitting of the MLP models. MLP models are one of the classic and fundamental neural network architectures used for solving classification or regression problems. The proposed approach builds upon the traditional MLP models by incorporating MC dropout, which generates multiple samples from the network with randomly dropped-out neurons during inference. The study showcases the versatility of the MLP model with MC dropout in various applications, including regression, time series forecasting, and classification tasks. The proposed approach outperforms the traditional MLP models in terms of accuracy and robustness. The study provides evidence that MLP models with MC dropout can be a useful tool in various domains where neural network models are utilized. In [Orr+20], the authors compare the performance of support vector machines (SVM) and Multi-Layer Perceptron (MLP) models for RUL prediction based on sensor readings from a real operating

centrifugal pump in the SARLUX refinery in Italy. Although the paper does not provide detailed information about the model configurations, it highlights the potential of using simple and easy-to-implement machine learning algorithms for forecasting the health status of monitored machines.

2.4 Long Short Term Memory

The article [NM19] proposes a novel approach for predicting the probabilities of RUL falling within different time windows using a Long Short-Term Memory (LSTM) neural network. This method is different from the previously mentioned studies, which focus on predicting the entire distribution of RUL. The LSTM network is trained to predict the probabilities of the RUL falling within different time windows, such as 10, 20, or 30-time units, for the turbofans in the C-MAPPS dataset. The proposed approach shows promising results, achieving high accuracy in predicting the probabilities of RUL falling within different time windows. This method has practical applications in real-world scenarios, where it is often more important to predict the probability of failure within a certain time window than the exact RUL. For example, in maintenance scheduling, it is more efficient to schedule maintenance based on the probability of failure within a certain time window rather than waiting for the exact RUL. In [Rem+22], the authors use a combination of Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BDLSTM) approaches to predict the RUL. Their approach involves using two CNN layers to extract feature maps, with each layer having ten feature maps. The resulting feature maps are then reduced to a single feature map, which is fed into the BDLSTM path. This path consists of two BDLSTM layers. The authors compare their method to other methods for the C-MAPPS dataset and show that their hybrid approach outperforms other methods that use Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), and BDLSTM. The authors use four subsets of the C-MAPPS dataset to demonstrate the effectiveness of their approach in predicting RUL. Their study highlights the potential benefits of combining CNN and BDLSTM approaches for RUL prediction, and their findings suggest that this hybrid approach can achieve better performance than other state-of-the-art methods. The current research on RUL prediction has explored a variety of approaches, ranging from basic machine learning algorithms that are easy to implement to more complex models that combine multiple algorithms. The studies also suggest that the C-MAPPS dataset is the most commonly used dataset for prognostic health management research. While most studies on RUL prediction focus on point predictions, there are some studies that use probability distributions to provide more detailed information about the RUL. These studies typically use Bayesian methods or other probabilistic models to estimate the uncertainty associated with the RUL prediction, and they can provide additional insights into the health of the monitored system. However, these approaches may also require more complex modeling and computational resources.

2.5 Mixture Density Network

Mixture Density Networks (MDNs) are a type of neural network that output a set of distributions, which results in a multimodal distribution. In [Nil+21] MDNs are used for the scheduling of automated radiation therapy. Where BNNs result in a single mean and standard deviation, MDNs utilize a set of mean values and standard deviations as well as the importance of each of the distributions. In [FHZ23] MDNs are used for the estimation of the SOH of Lithium-Ion batteries with uncertainty measurements. In [Kim+23] the use of an MDN is used for the prediction of the wear of tools under different machine conditions. In [Che+22] the degradation of battery capacity is predicted using a MDN struture, where the RUL and SOH are predicted.

3 Data Description - batteries for eVTOLS

In the context of this research focused on eVTOLS, the primary dataset utilized will be the Vahana dataset. This dataset includes comprehensive information concerning the Lithium-Ion batteries employed within the Vahana eVTOLS framework. Within this dataset, an array of distinct mission profiles is incorporated. These mission profiles are essentially composed of a series of interconnected missions, with each profile maintaining consistent specifications throughout. To elucidate the concept of a mission, it can be deconstructed into a succession of distinct phases. Each mission initiates with two consecutive charging phases. The initial phase is characterized as the Constant Current (CC) charging phase, during which the battery receives a continuous current for recharging purposes. Subsequently, the second charging phase, referred to as the Constant Voltage (CV) charging phase. In contrast to the CC charging phase, the CV charging phase employs a stable voltage for battery recharging. Following these charging phases, a resting phase occurs, followed by the flight phases. The flight phases are sequenced as follows: Take-Off, Cruise, and Landing. After the completion of the landing phase, a concluding rest phase is initiated. After this overview, the subsequent chapter will delve into a more intricate exploration of the dataset's particulars.

3.1 Mission Profile

The Vahana dataset is comprised of a series of files, each containing data related to multiple flight instances. Within these files, specific mission profiles can be specified, serving to define various variables throughout a mission. Notably, three mission profiles, namely VAH01, VAH17, and VAH27, are established as baseline mission profiles. These baseline mission profiles are outlined as follows: During the CC charging phase, the battery undergoes charging at a rate of 1-C. This phase of CC charging continuous until the battery voltage reaches 4.2V. Following this, the CV charging phase begins, maintaining a constant voltage of 4.2V until the current drops below C/30. It's important to note that the battery temperature experiences an increase during the charging process. After charging, a cooling period ensues, concluding when the battery temperature reaches 35 degrees Celsius. Following the resting period, the flight phase begins. Within the baseline mission profile, the flight is characterized as follows: The take-off phase spans 75 seconds, during which a discharge rate of 5-C is observed. Subsequently, a cruise phase follows, spanning 800 seconds and a discharge of 1.48-C. The flight concludes with a landing phase, lasting 105 seconds and featuring a discharge rate of 5-C.

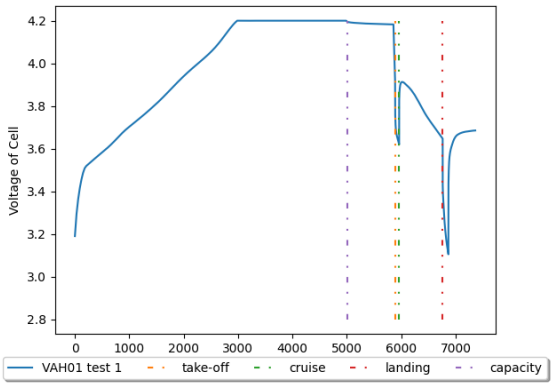
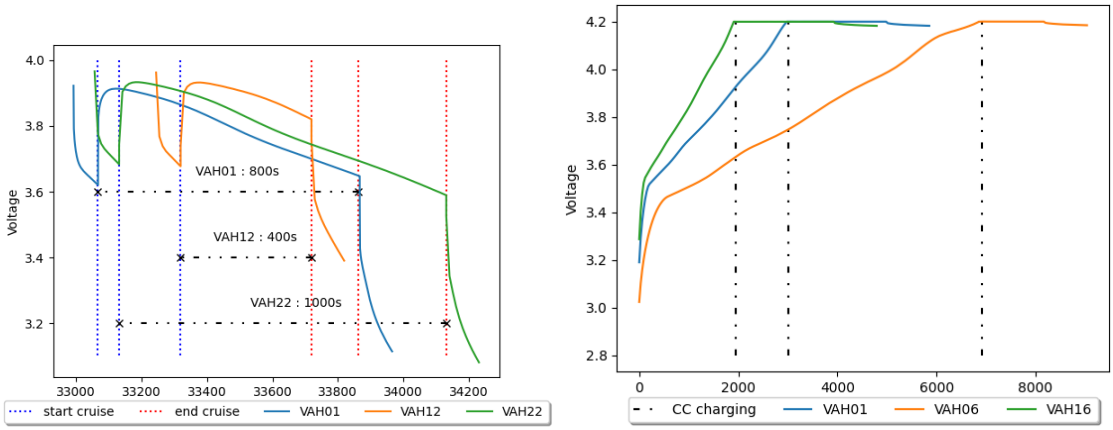


Figure 1: Battery Voltage of first Capacity test of VAH01

3.1.1 Contrasting Mission Profiles

As detailed in section 3.1, the datasets VAH01, VAH17, and VAH27 contain the baseline specifications. However, for the remaining files, certain variables within the mission profile are modified. Table 31 provides a comprehensive overview of these modifications within the mission profiles. As illustrated in the table, distinct variations are introduced across each mission profile. These deviations include diverse aspects, including cruise duration, CC current rate, flight power consumption, CV voltage, and ambient temperature. In figure 1 the voltage of the battery cell for the first capacity test is visualised with the beginnings each of the flight phases.



(a) First flight of VAH01, VAH12 and VAH22

(b) CC-rate differences

Figure 2: Contrasting Mission Profile examples

The diverse alterations introduced exhibit varying impacts on battery deterioration. Figure 2 graphically represents the effects of changes in cruise time and CC-rate. Figure 2a showcases the initial flights of VAH01 (baseline), VAH12, and VAH22. A prominent observation is the comparison of cruise times; while VAH01 adheres to the baseline mission profile cruise duration of 800 seconds, VAH12 decreases its cruise duration to 400 seconds, and VAH22 increases the cruise duration to 1000 seconds. Evidently, the prolonged cruise duration directly correlates with a substantial decline in voltage. This decline becomes more pronounced with increasing cruise duration, signifying a noteworthy correlation between the two variables. An additional significant alteration relates to the CC-rate. Illustrated in Figure 2b, the graph clarifies differences in CC charging durations. Where VAH06 has a CC rate of 0.5-C and VAH16 has a CC rate of 1.5-C. The increase in CC rate shortens the duration of the CC charging duration and a decrease in CC rate extends the CC charging duration.

3.2 Data Specifications

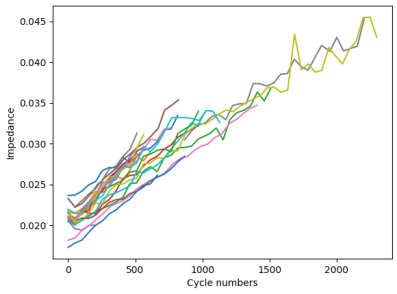
As we have seen before the data consist of multiple mission profiles with different alternations, where each mission profile is stored in a separate file. The data consist of the following measurements: Time stamp; Cell voltage; Cell current; The quantity of energy transferred during the charging process, expressed in watt-hours(Wh); The quantity of energy transferred during charging, measured in milliampere-hours (mAh); The

energy expended during discharge, denoted in watt-hours; The energy utilized during discharge, represented in milliampere-hours (mAh); Surface temperature of the cell, measured in degrees Celsius (°C) Cycle number; and Cycle segment.

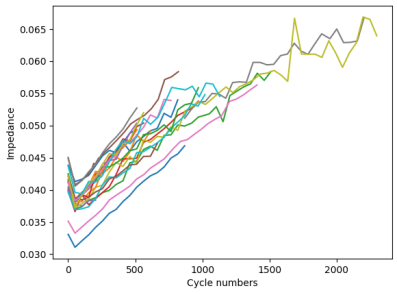
3.3 Impedance Data

Previous research using the eVTOL battery dataset focused on the data mentioned in section 3.2. Since April 2023 the data files containing the impedance of the battery are released. The impedance can be explained as: the overall resistance that a battery presents to a flow of AC. The Depth of Discharge (DOD) is the percentage of the remaining charge compared to a fully charged battery. The 1 and 30 seconds refer to a relax time where the battery is at a current of $C/50$. The following formula measures the impedance: $\frac{|\text{voltage}_{\text{before}} - \text{voltage}_{\text{after}}|}{|\text{current}_{\text{before}} - \text{current}_{\text{after}}|}$ The impedance data for each of the mission profiles are in the following order:

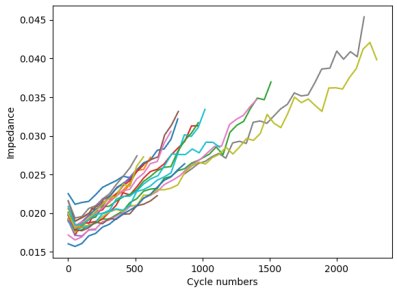
- $I_{20\%,1}$: Impedance at 20% DOD after 1 second of low current
- $I_{20\%,30}$: Impedance at 20% DOD after 30 seconds of low current
- $I_{60\%,1}$: Impedance at 60% DOD after 1 second of low current
- $I_{60\%,30}$: Impedance at 60% DOD after 30 seconds of low current



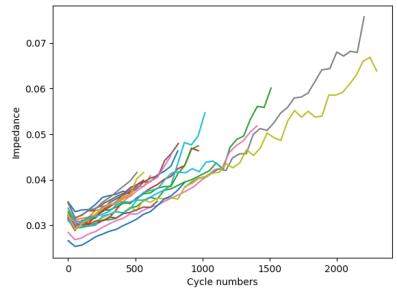
(a) 20% 1 second



(b) 20% 30 seconds



(c) 60% 1 second



(d) 60% 30 seconds

Figure 3: Impedance data

The impedance of a battery will increase over time. The reason for the increase in impedance over time is due to the chemical reactions inside the battery. The chemical reaction between electrolytes and electrodes inside the battery has as a result the decrease of power the battery can deliver. This chemical reaction slows down over time due to the deterioration of the electrodes. In figure 3 each of the measurements are displayed. Here we can clearly see that the impedance increases over-time with linear characteristics.

3.4 Capacity Tests

Having outlined the sequence of mission phases, comprehended mission profiles, and defined data specifications, we now introduce the concept of capacity tests. These tests serve the purpose of assessing the battery's capacity and will be executed following every 50 missions. The capacity test procedure initiates with discharging the battery at a C/5 rate. This discharge continues until the battery's voltage reaches 2.5V and its state of charge (SOC) reaches 0%. Subsequently, a period of rest is observed, during which the battery's temperature must drop below 30 °C. Following this cooling phase, the charging cycles are initiated.

Capacity test	CC duration	CV duration	Rest period
1	3012s	2004s	870s
5	2695s	2453s	870s
9	2510s	3005s	870s
12	2418s	3262s	870s

Table 3: CC and CV duration for capacity tests of VAH01

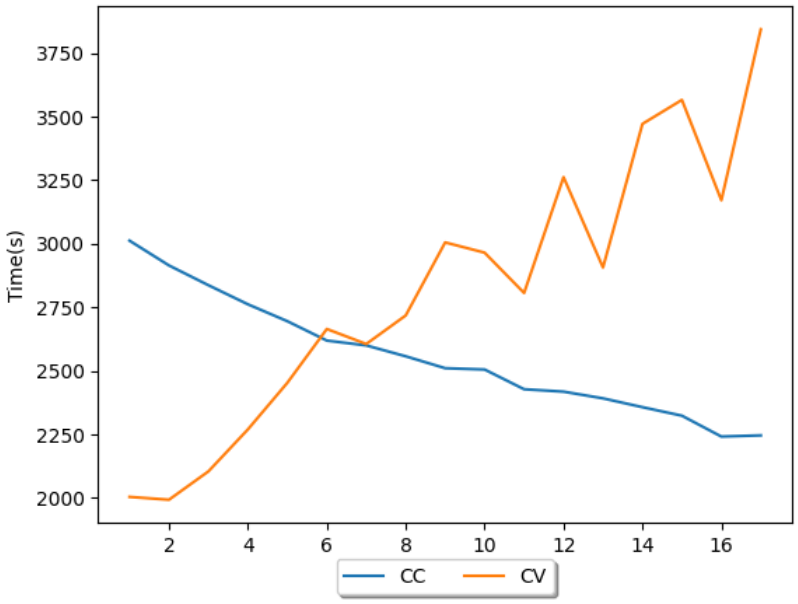


Figure 4: CC and CV duration of VAH01

Figure 4 portrays the CC and CV durations of all capacity tests conducted in VAH01. Within VAH01's CC charging phases, a consistent 3A current is used. Given the 1-C charging rate for the CC charging phase, we determined $C = 3A$. The first capacity test has a CC duration of 3012 seconds, while the last capacity test has a CC duration of 2246 seconds. Evidently, this denotes a reduction of 766 seconds in the CC duration. Notably, the CC duration displays a diminishing trend with increased battery use, whereas the CV duration demonstrates an opposite pattern. Commencing with a CV time of 2004 seconds in the first capacity test, the CV duration increases to 3844 seconds in the final test, illustrating an increase of 1840 seconds. At the end of the CV charging phase the state of charge is 100%. At this point the remaining total capacity of the battery is measured. Following the charging phases, the initial rest interval takes place until the battery's temperature reaches $35°C$. Notably, across all capacity tests in VAH01, a uniform rest period of 870 seconds is observed. After the rest period, the take-off flighing phase begins. As discussed in section 3.1, this phase involves a discharge rate of 5-C. This discharge is clearly visible in Figure 1, where the start of the take-off is indicated by an orange dotted line. After the take-off, the cruise phase follows, which uses less energy at a rate of 1.48-C. During the landing phase, the energy consumption increases again to 5-C. The reason for the higher energy consumption during take-off and landing, compared to the cruise phase, is that the eVTOL needs more energy to change its altitude, which requires more power than maintaining a steady height.

3.4.1 Definitions of SOH, RUL and RUC

To assess the values for the prognostic predictions we will be using SOH, RUL and RUC. The SOH(State of Health) can be defined as:

$$SOH^{m,cc} = \frac{Capacity^{m,cc}}{Capacity^{m,0}} * 100\% \tag{1}$$

where m is the mission profile and cc is the capacity test for which the SOH will be determined. The RUL(Remaining Useful Life) is the number of remaining flights until the Battery hits the EOL(End of Life). The RUL can be defined as:

$$RUL^{m,cc} = T^{m,EOL} - T^{m,c} \tag{2}$$

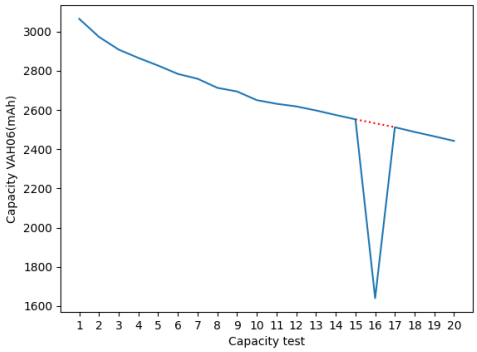
where m is the is the mission profile and cc is the current capacity test. $T^{m,EOL}$ is the first flight where the battery reaches the EOL. The RUC(Remaining Useful Capacity tests) is a representation of the RUL, where the RUC is the remaining capacity tests before the EOL is hit. The RUC can be defined as:

$$RUC^{m,cc} = \frac{RUL^{m,cc}}{51} \tag{3}$$

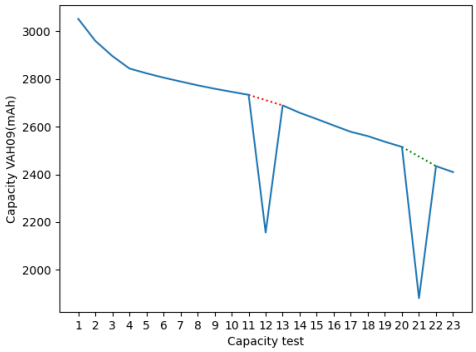
Where m is the mission profile and cc the current capacity test.

3.5 Mission Selection

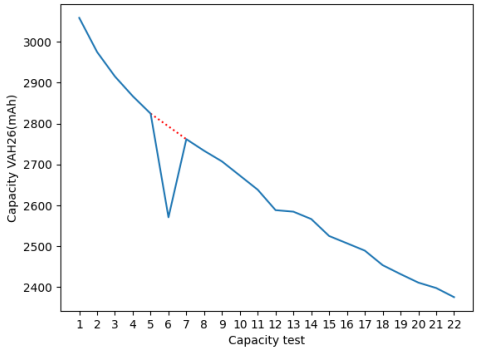
In our predictive analysis, we'll utilize all the profiles available. However, upon data examination, certain mission profiles pose some issues. In Figure 5, we can see the capacities measured for VAH06, VAH09, VAH26, and VAH27. Normally, we would expect the battery capacities to decrease gradually over time. However, in this figure, these mission profiles exhibit capacity tests where the capacity unexpectedly decreases compared to the successive. Consider VAH06 as an example. In its 16^{th} capacity test, the capacity has dropped to 1641 mAh, whereas the test before measured a capacity of 2553 mAh, and the successive is 2512 mAh. Likewise, VAH09 displays capacity tests that deviate from the expected pattern.



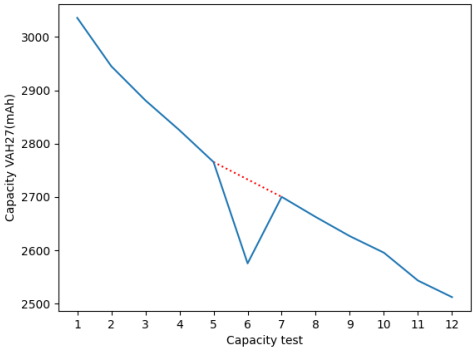
(a) VAH06



(b) VAH09



(c) VAH26



(d) VAH27

Figure 5: Capacities of VAH06, VAH09, VAH26 and VAH27

In the 12^{th} capacity test, there is a decline in capacity to 2156 mAh, whereas the successive test records a capacity of 2689 mAh. The second issue occurs in the 20^{th} capacity test, where the capacity drops to 0 mAh. This occurs due to alternating data, where the charge delivered in every other interval is incorrectly labeled as discharge. The same issue is observed in both VAH26 and VAH27. Where capacity reads lower than the successive capacity test.

To rectify these inaccuracies in the mission profiles, we will apply the following resolution method:

$$capacity_i = \text{mean}(Capacity_{i-1}, Capacity_{i+1})$$

Here, i represents the capacity test of irregular behavior. The red dotted lines in figure 5 represents the capacities after rectification.

3.5.1 VAH07

Mission profile VAH07 consist of only 5 capacity tests. The last capacity test has a SOH of 91.56%, therefore the EOL is not reached. This indicates that the data within VAH07 cannot be utilized for predicting the RUL, as the RUL cannot be reliably determined. This means that mission profile VAH07 only will be usable for the SOH prognostic predictions. As a result, the data from mission profile VAH07 can only be used to predict the SOH. This limitation stems from the fact that RUL prediction requires a scenario where the EOL is reached, which is not witnessed.

3.5.2 Missing data

For VAH11 and VAH26 the impedance data is missing or incomplete. For VAH11 the impedance data only starts from cycle number: 868, this has the problem that for the first 17 capacity tests the impedance data is missing. when it comes to VAH26 all the impedance data is missing, due to the absence of a file for VAH26.

4 Methodology for data-driven prognostics of SOH, RUL and RUC

In this section we will be defining the methodology for the prediction of the prognostic distributions for the SOH, RUL and RUC. First we will be discussing the features retrieved and engineered from the dataset discussed in section 3. Next, we will introduce a framework for predicting prognostic distributions, implementing various approaches. This framework consists of a multi-step process including: feature extraction, feature importance, proposed algorithms, hypertuning, cross-validation process, preprocessing and evaluation metrics.

4.1 Features

From the data described in section 3, we can derive a set of features. Specifically, the features outlined in [Mit+23] demonstrate an understanding of the diverse types of features applicable to the non-impedance data. These features encompass a comprehensive set aimed at characterizing the data. All features will be conducted for each of the mission profiles m and each of the capacity tests cc. In addition to these features we will be exploring the impedance data as a set of added features.

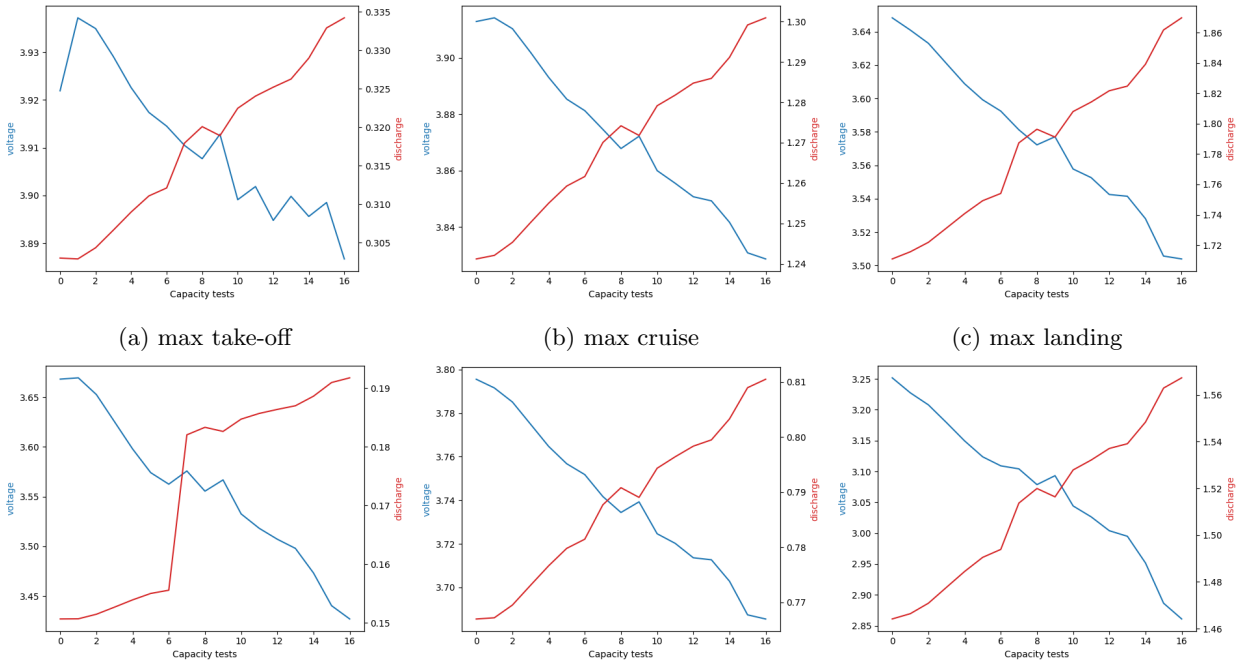
Charging Features

The first features we will discuss are the features related to the charging phases. As we mentioned before the charging phases consist of a CC charging phase where the current has a constant rate and a CV charging phase where the voltage will be in a constant. After the CC charging and CV charging phases a rest phase take place. The features that are introduced are features containing the duration of the CC, CV and rest phase. This will create a set of three features: $delta_{CC}^{m,cc}$, $delta_{CV}^{m,cc}$ and $delta_{REST}^{m,cc}$.

Discharging Features

Moving on to the features associated with the discharging of the flight phases, which contains take-off, cruise, and landing. In Figure 6, the maximum and mean voltage, along with the discharge values during

each of these discharge phases, are displayed. The figure demonstrates a noticeable decline in voltage as time progresses, accompanied by an increase in discharge.



(a) max take-off (b) max cruise (c) max landing

(d) mean take-off (e) mean cruise (f) mean landing

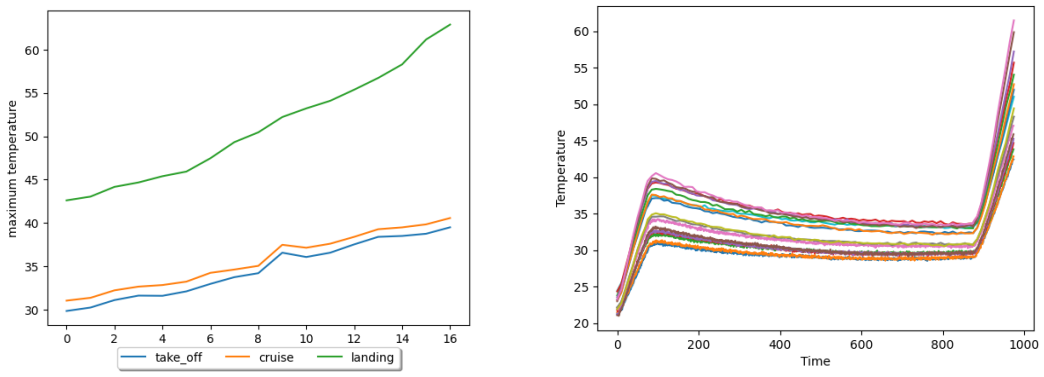
Figure 6: discharge voltage(blue) and capacity(red) of VAH01

We can see these characteristics in relation to the deterioration of the batteries. The lowering of the voltage over time can be explained with regard to the impedance of the batteries. The discharge capacity and its variations mirror the load characteristics of the battery, exerting a direct influence on the battery's aging process. This introduces the following features for each of the flighting phases:

- $V_{max}^{phase,m,c}$, $V_{min}^{phase,m,c}$, $V_{mean}^{phase,m,c}$ and $V_{var}^{phase,m,c}$
- $Qdis_{max}^{phase,m,c}$, $Qdis_{min}^{phase,m,c}$, $Qdis_{mean}^{phase,m,c}$ and $Qdis_{var}^{phase,m,c}$

Temperature Features

As the battery ages, the average temperature of its cell surfaces tend to rise. This trend is illustrated in Figure 7a, which displays the maximum temperature of the VAH01 battery during its discharge phases.



(a) max temperature

(b) temperature over time

Figure 7: Temperatures of VAH01 discharge phases

Figure 7a shows an increase in temperature for all discharge phases and figure 7b show that the take-off and landing have the most pronouncing increase in temperature. The increase can be explained by the fact that the take-off and landing have a larger discharge rate compared to the cruise phase. This introduces three features: $T_{max}^{phase,m,cc}$

Impedance features

As outlined in section 3.3, a recently released dataset contains impedance measurements from capacity tests of the Vahana datasets. These measurements contain four distinct values, each showing a notable increase in impedance. Since each measurement corresponds to a different point, we will utilize these measurements as features. This introduces the following features: $I_{20\%,1}^{m,cc}$, $I_{20\%,30}^{m,cc}$, $I_{60\%,1}^{m,cc}$ and $I_{60\%,30}^{m,cc}$

4.2 Framework for prognostics of SOH, RUL and RUC using machine learning

For the Prognostic distribution prediction we developed a general frame work. The framework consists of a multi-step process for prognostic distribution. The goal of this framework is to create a uniform frame for implementing different types of prognostic distribution predictors. The first part of this frame work is the extraction of the features, which we discussed previous in section 4.1. The feature extraction consist of a two step method. First we will be retrieving the indexes of each of the moments:

- start of CC
- start of CV
- start of rest period
- start of take-off
- start of cruise
- start of landing

After retrieving all the indexes the computation of the features can occur, as mentioned in 4.1. For each of the indexes we can compute the features, importantly is the moment where the CV charging phase is finalised. At this point for each of the capacity test the current maximum capacity will be measured. Finally the flight phases will occur and the features of the take-off, cruise and landing will be extracted.

4.2.1 Feature importance

Before the predictions for the prognostic distribution can be executed, we need to determine the importance of each of the features. In total the feature extraction computes 33 features, of which we will be using the best 65% of features. For the feature importance we will be using the SHAP(SHapley Addictive exPlanations) values. These values are assigned to each of the features, whereas these values represent the level of contribution to the output of any machine learning algorithm. To compute the SHAP values we will be utilizing a Random forest model for which we will be computing the Shap values. Since we have three different prediction goals, we will be computing the importance for each of these values. In Appendix 10.2 the SHAP values for each of the goal values are shown. As mentioned in section 3.4 the CC and CV phases decrease and increase respectfully. The SHAP values show the importance of features where over time a clear increasement or decreasement is visible. Especially the features about the voltage during take-off are of high importance. The impedance based features are especially of high importance for the prediction of SOH.

4.2.2 Predicting the distribution of SOH, RUL, RUC - probabilistic prognostics

In the context of prognostic distribution predictions, we will examine two distinct algorithmic approaches. The first method entails employing Monte Carlo dropout both during training and prediction phases. In contrast, the second approach involves predicting the mean and standard deviation parameters to characterize a normal distribution.

MC dropout

We will explore two distinct algorithms for implementing dropout in our study. The first algorithm employs CNN with MC dropout applied to the hidden layers. The second algorithm utilizes LSTM networks with MC dropout. Traditionally, dropout is employed to prevent overfitting and enhance generalization. In our case, for estimating prognostic distributions, each input will undergo multiple sampling instances during prediction, with dropout enabled during estimation.

Normal distribution

The second category of algorithms we will explore operates on the principle of estimating the mean and standard deviation of each of the prognostic distribution. These methods utilize the negative likelihood as the loss function, which inversely represents the likelihood function, used to maximize the probability of observation undergoing the given models. In training, the goal is to minimize this negative likelihood function. We will consider four distinct algorithms for this purpose. The first two algorithms involve CNNs and LSTMs, where the output comprises the mean and standard deviation. The third algorithm is a BNN, which differs from CNNs or LSTMs by incorporating distributions as weights in its layers, rather than conventional values. The objective of a BNN is to integrate Bayesian principles into neural network architecture, resulting in a model that not only predicts output but also accounts for prediction uncertainty.

Figure 8 illustrates the conceptual idea behind the algorithms that yield the mean and standard deviation of the normal distributions.

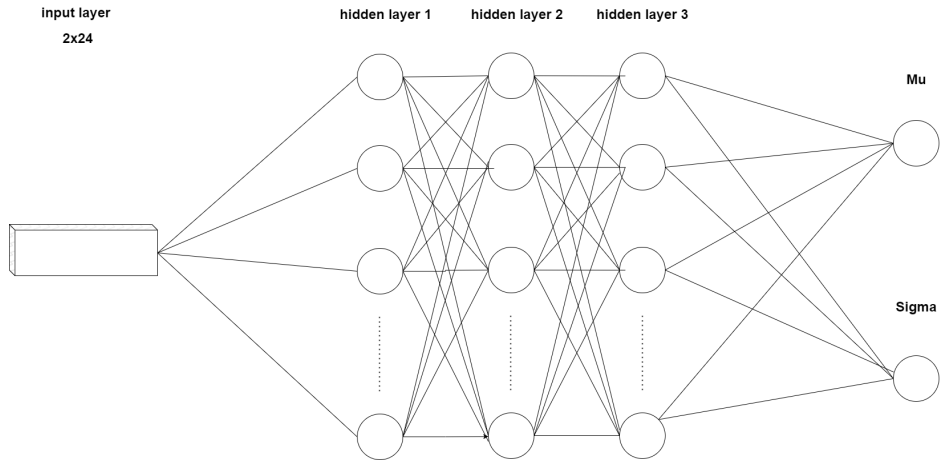


Figure 8: NN normal distribution Scheme

In the depicted architecture, the input layer is configured with dimensions of 2x24, followed by a variable number of hidden layers. These hidden layers may consist of CNNs, LSTMs, or BNNs, depending on the chosen algorithm. Regardless of the specific algorithm, all models share a common output layer responsible for generating the Mu (mean) and Sigma (standard deviation) values. In addition to the aforementioned algorithms, we will also examine a MDN. While the previous algorithms produce outputs conforming to a normal distribution, the MDN is designed to incorporate multiple normal distributions. The output of the MDN comprises multiple means and standard deviations, each corresponding to distinct distributions. Moreover, weights are assigned to each distribution to indicate its relative importance in the prediction process.

4.2.3 Leave one out - Cross Validation

To validate the performance of each of the algorithms we will be utilizing the leave-one-out cross validation method, where cross validation is used where each mission profile will serve as a validation set, and the remaining of the mission profiles will be the training set. Consequently, we will have 21 distinct validation sets for the SOH and 20 for the RUL and RUC respectively, across all algorithms. For each of these validation sets, the evaluation process will consist of three main steps: hyperparameter tuning, training, and evaluation. Initially, hyperparameter tuning will be conducted to optimize the parameters of the algorithms. Subsequently, the algorithms will be trained on the training data, utilizing the optimized hyperparameters. Finally, the trained models will be tested and evaluated on the respective validation sets to assess their performance. This comprehensive evaluation process ensures that each algorithm is rigorously assessed across multiple validation sets, providing a robust understanding of its effectiveness in predicting SOH, RUL, and RUC.

4.2.4 Preproccesing

For preprocessing the data before inputting it into each algorithm needs to be reshaped, we will merge two capacity tests into a single input. Specifically, we will select the 24 most important features from each capacity test and combine them into a single input. This transformation will result in a data format of 24 features by 2 capacity tests. thereafter, for each of these input instances, we will utilize the output from the second capacity test as the target output for training and evaluation. For instance, if we were to forecast the SOH of the initial preprocessed input from VAH01, it would comprise data from the first and second capacity tests, with the intended prediction focusing on the SOH during the second capacity test. This preprocessing step ensures that each algorithm receives consistent input data format and enables the utilization of pertinent information from multiple capacity tests to enhance predictive accuracy.

4.2.5 Hyperparameter tuning

After preprocessing, the next step involves hyperparameter tuning, for which we'll utilize the Keras Tuner library. This library offers various methods for hyperparameter tuning, each with its own characteristics. One such method is the GridSearch tuner, which extensively explores all possible combinations of hyperparameters. While effective, this approach becomes computationally expensive when dealing with large search spaces. For instance, tuning a simple neural network with 1 to 5 layers and 50 possible neurons per layer would result in more then a billion possible combinations. Another approach is the RandomSearch tuner, which randomly samples the search space a certain number of times to find the best combination of hyperparameters. If the number of samples used equals the maximum possible combinations, RandomSearch effectively functions as GridSearch. However, the method we'll utilize for hyperparameter tuning is Bayesian Optimization. This technique utilizes Bayesian probabilistic methods to efficiently search for the optimal set of hyperparameters in as few iterations as possible. In this optimization process, there remains a constraint on the number of trials that must be specified. As the Bayesian optimizer identifies a set of hyperparameters close to the optimal configuration, it will approach the lower bound. This indicates that the algorithm has start reaching a point where no further significant improvement can be achieved. Notably, Bayesian Optimization requires setting certain parameters before initiation, namely the α and β values. Determining suitable values for α and β typically involves a trial-and-error process. By leveraging Bayesian Optimization, we aim to streamline the hyperparameter tuning process while ensuring optimal performance of the algorithms.

- α : The expected amount of noice in the observed performance of the Bayesian optimizer(default: 0.0001)
- β : The factor that balances exploration and exploitation, whereas the higher the value the greater the exploration(default: 2.6)

For hyperparameter tuning across the various algorithms, we will rely on the BayesianOptimizer due to its ability to balance resource utilization with performance effectively. When configuring the BayesianOptimizer, the choice of α and β values is crucial, with particular emphasis on the β value. This parameter determines the degree of greediness manifested by the optimization process as it traverses the search space. As represented by Grid Search, even a modestly complex scenario involving a fully connected neural network with 1 to 5 layers and 50 potential neurons per layer results in an expansive search space of more then a billion options. This example underscores the exponential growth of the search space as algorithm complexity increases. Therefore, it's imperative to wisely select hyperparameters to balance computational resources and performance effectively. Furthermore, each algorithm will necessitate exploration of a distinct set of

hyperparameters, reflecting the unique characteristics and requirements of each model. By tailoring the hyperparameter search to the specific needs of each algorithm, we aim to maximize performance while mitigating computational overhead.

CNN

First we will be exploring the hyperperameters for the CNN using Mc dropout. Here we have the following hyperparameters:

- Number of **convolutional layers**.
- Number of **kernals** for each layer.
- Size of these **kernals** for each layer.
- Type of **activation** function used for each layer.(**tanh**, **relu**)
- Number of **units** in the last **dense** layer.
- Type of **activation** function for dense layer.(**tanh**, **relu**)

In addition to hyperparameters, several other values must be determined for hyperparameter tuning, including the α and β values discussed earlier, as well as the number of trials and epochs. By default, we will use the default values for α and β. The number of trials and epochs during hyperparameter tuning plays a crucial role in the optimization process. Increasing the number of trials provides a broader exploration of the search space, aiding in better understanding the optimal hyperparameters. However, there exists a point of diminishing returns, where additional trials may not significantly improve optimization outcomes. Similarly, increasing the number of epochs during hyperparameter tuning enhances the model's training duration, potentially leading to improved performance. However, this also incurs escalating computational costs until reaching a point of diminishing returns. For CNN models where the output comprises the Mean and Standard Deviation, the hyperparameter tuning process largely remains consistent. However, differences may arise in the final layer, particularly with the specialized **independentnormal** layer. Notably, although the types of hyperparameters for CNNs with MC dropout and CNNs with normal distribution may appear similar, it's essential to recognize that their tuned parameters may differ due to the distinct nature of their outputs.

LSTM

Next, we delve into the essential hyperparameters for the LSTM models. Similar to the CNN implementations, both the LSTM model employing MC dropout and the one utilizing a Normal distribution as output follow a similar principle. However, it's crucial to note that the hyperparameters for LSTM models differ from those of CNNs. The following hyperparameters require tuning for the LSTM models:

- Number of **LSTM layers**.
- Number of **LSTM units** in each layer.
- Type of activation function for each layer: (**tanh**, **relu**, **sigmoid**)
- Number of **units** in the last **dense** layer.
- Type of activation function for the last layer: (**tanh**, **relu**, **sigmoid**)

BNN

In the context of the BNN, specialized layers incorporate units utilizing prior and posterior distributions. The hyperparameters for tuning include:

- Number of **bayesian layers**.
- Number of **units** for each layer
- Type of Activation function: (**tanh**, **relu**, **sigmoid**).
- Number of **units** in the last **dense** layer.
- Type of activation function for the last layer: (**tanh**, **relu**, **sigmoid**)
- kl-weights
- kl-use exact

As evident from the hyperparameter set, besides the typical parameters like the number of layers and units in each layer, more intriguingly, the kl-weight and kl-use-exact parameters are also subject to hypertuning. The kl-weight parameter serves as the scale for the kl divergence loss between the prior and posterior distributions. On the other hand, the kl-use-exact value determines whether the kl divergence will be employed instead of Monte Carlo approximation.

MDN

In contrast to the previously mentioned algorithms, the shape of the output is also determined during hyperparameter tuning. therefore, for the MDN, the number of normal distributions that will be outputted is subject to hypertuning. For the MDN, the following hyperparameters will be considered:

- Number of normal distributions in the output
- Number of hidden dense layers
- Number of units in each hidden dense layer
- Activation function in each of the dense layers.

4.2.6 Evaluation Metrics

To evaluate the algorithms we will be using a set of metric scores to evaluate the performance. We can define the metrics in to sets. The first two metrics are the MAE and RMSE scores. These scores use the mean value of the predicted prognostic distributions, comparing to the actual values. Both MAE and RMSE are defined as followed:

The Mean Absolute Error (MAE) is defined as:

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4}$$

The Root Mean Squared Error (RMSE) is defined as:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{5}$$

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

novel metrics

The second set of metrics that will be utilized are the novel metrics mentioned in section 2.1. These metrics define aspects of the distribution that will be lost when using just the mean of the distribution for evaluation. The first novel metric that will be used is the CRPS score which gives a score for the accuracy and sharpness of the distributions. The CRPS can be defined as follows:

$$CRPS = \frac{1}{N}\sum_{i=1}^{N} CRPS_i \tag{6}$$

$$CRPS_i = \int_{-\infty}^{\infty} (F_{\hat{y}_i}(x) - \mathcal{I}\{y_i \leq x\})^2 dx \tag{7}$$

$$with\ \mathcal{I}(\alpha)_i = \begin{cases} 1, & y_i \leq x \\ 0, & y_i > x \end{cases} \tag{8}$$

As seen in previous studies about PHM systems overestimating is more harmfull compared to underestimating. Since overestimation can lead to severe problems and dangerous situations and failures. To address this, the weighted version of the CRPS score is used, which penalizes the overestimation of predictions using a user-defined parameter, $0 \leq \beta \leq 2$. The weighted CRPS score is defined as follows:

$$CRPS^W = \frac{1}{N}\sum_{i=1}^{N} CRPS_i^W \tag{9}$$

$$CRPS_i^W = (2-\beta)\int_{-\infty}^{y_i} (F_{\hat{y}_i}(x) - \mathcal{I}\{y_i \leq x\})^2 dx + \beta \int_{y_i}^{\infty} (F_{\hat{y}_i}(x) - \mathcal{I}\{y_i \leq x\})^2 dx \tag{10}$$

While the CRPS score covers accuracy and sharpness, reliability can be measured using the α-Coverage metric introduced in [PM22]. The α-Coverage is defined as follows:

$$\alpha - Coverage = \frac{1}{N}\sum_{i=1}^{N} \mathcal{I}(\alpha)_i, \tag{11}$$

$$with\ \mathcal{I}(\alpha)_i = \begin{cases} 1, & y_i \in [\hat{y}_i^{0.5-0.5\alpha}, \hat{y}_i^{0.5+0.5\alpha}] \\ 0, & otherwise \end{cases} \tag{12}$$

Here, the level of uncertainty is deemed to be overestimated if α percentage of all the test instances are within the interval, otherwise, it is considered to be underestimated [PM22]. The last metric introduced is the RS-score, which is different from the α-coverage as it evaluates the uncertainty across all α values.

The RS-score is computed by using the reliability curve, which is the difference between a certain α and the width of the credible interval. The RS-score is then computed by measuring the area between the reliability curve and the ideal curve. The ideal curve is defined as the curve where the coverage is equal to α. The RS-score is defined as follows:

$$RS^{under} = \int_0^1 \mathcal{I}\{C(\alpha) \leq \alpha\}(\alpha - C(\alpha))d\alpha \tag{13}$$

$$RS^{over} = \int_0^1 (1 - \mathcal{I}\{C(\alpha) \leq \alpha\})(C(\alpha) - \alpha)d\alpha \tag{14}$$

$$RS^{total} = RS^{under} + RSover \tag{15}$$

$$with\ \mathcal{I}\{C(\alpha) \leq \alpha\} = \begin{cases} 1, & C(\alpha) \leq \alpha \\ 0, & Otherwise \end{cases} \tag{16}$$

Combining the generic scores such as the MAE and the RMSE and the different novel metrics we can get a better and having more meaningfull understanding of performance of the predicted distributions.

5 Overall results

In this section, we will provide an overview of the results for SOH, RUL, and RUC , which will be presented in separate sections (6, 7, and 8) in more detail. We will evaluate the overall performance of all the algorithms using the MAE from the general metrics, which gives an initial representation of the performance of the algorithms. Additionally, we will consider the CRPS, RS under and RS over for a better understanding of the prognostic distributions. For SOH predictions, the LSTM with MC dropout emerged as the best-performing algorithm based on MAE. However, when considering CRPS and RS under, the MDN exhibited superior performance. Moving on to RUL predictions, the MDN showes strong performance in terms of MAE. However, when examining CRPS and RS under, both the CNN with MC dropout and LSTM with mean stddev output demonstrated competitive performance. Lastly, for RUC predictions, the LSTM with mean stddev output performed the best based on CRPS. Overall, while certain algorithms performed better in terms of MAE, considering additional metrics such as CRPS and RS under provided a more comprehensive understanding of their predictive capabilities. It is essential to assess algorithm performance across various metrics to make informed decisions about their suitability for specific applications.

Algorithms		**MAE**			**CRPS**		
		SOH	*RUL*	*RUC*	*SOH*	*RUL*	*RUC*
CNN	*dropout*	1.58	70.10	1.33	1.443	64.093	1.243
	mean	2.37	79.36	1.60	2.760	62.402	1.284
LSTM	*dropout*	1.29	81.23	1.31	1.172	80.503	1.319
	mean	1.86	69.11	1.42	2.381	62.108	1.180
MDN		1.39	66.66	1.44	1.088	62.881	1.377
BNN		4.56	161.72	2.08	3.130	114.403	1.586

Table 4: Overview of metrics for the SOH, RUL and RUC

Algorithms		rs under			rs over		
		SOH	*RUL*	*RUC*	*SOH*	*RUL*	*RUC*
CNN	*dropout*	0.368	0.345	0.378	0.000	0.000	0.000
	mean	0.043	0.049	0.153	0.005	0.000	0.000
LSTM	*dropout*	0.423	0.453	0.458	0.000	0.000	0.000
	mean	0.057	0.171	0.257	0.006	0.000	0.000
MDN		0.280	0.382	0.378	0.000	0.000	0.000
BNN		0.001	0.045	0.036	0.035	0.002	0.034

Table 5: RS scores of for the SOH, RUL and RUC

In Table 4 and 5, an overview of the metrics is provided, with green cells indicating the best-performing algorithm and yellow indicating the second best performing. Based on the MAE, the LSTM with MC dropout emerges as the best-performing algorithm, as it performs best for both SOH and RUC predictions. The MDN follows closely, performing second best for SOH and best for RUL predictions. Comparing MAE and CRPS, the LSTM with mean stddev output emerges as the best-performing algorithm. As depicted in Table 4, the LSTM based algorithms are the overall best-performing algorithms when considering both MAE and CRPS. Additionally, considering the RS under score provides insight into the underestimation tendencies of each algorithm. The LSTM with MC dropout exhibits the highest rs under value, indicating a greater degree of underestimation across SOH, RUL, and RUC predictions. Understanding under and overestimation tendencies is crucial for PHM systems. Notably, the BNN is the only algorithm showing a higher overestimation than underestimation, highlighting its unique behavior.

6 Estimating the distribution of SOH (probabilistic prognostics)

In this section we introduce the results for the prognostic distribution prediction of the SOH in more detail. For all algorithms we will be using the 24 most important features as we mentioned in section 4.2.1. These importance's are computed using the shap values. After the preprocessing using the shap valued importances the hypertuning the algorithms will be done to determine optimized hyper parameters for each of the algorithms. After hypertuning each algorithm using the hypertuned parameters will be fitted on the training data. Finally we will predict the prognostic distributions and compute to sets of metrics for the evaluation of the algorithms.

6.1 Hypertuning results SOH

In the hyperparameter tuning process for the SOH, each fold in our leave-one-out approach required individual hyperparameter tuning. This meticulous approach ensures a fair and consistent evaluation of the performance of each algorithm. In many computational-heavy machine learning problems necessitating hyperparameter tuning, it's customary to conduct tuning on a small subset of the data. To maintain fairness and consistency, we performed hyperparameter tuning for each fold in the leave-one-out approach. Given that we utilize the leave-one-out approach for each algorithm, this implies that we conduct hyperparameter tuning 20 times for SOH and 19 times for RUL and RUC. Consequently, the hyperparameter tuning will vary for each fold, yielding a set of hyperparameters. We present a general configuration for each algorithm.'It is important to mention the hypertuning will result in an optimized set of hyperparameters that is optimized

within the search space given a number of trials. When recomputing the hyperparameters a other set of optimized hyperparameters can be selected, given random seeds. Since the goal of hypertuning is finding a set of hyperparameters that work well given the training data. For each algorithm, we established a fixed number of trials during hyperparameter tuning, specifically 150. Each of these trials will conduct 60 epochs. These values will be the same for the SOH, RUL and RUC. This approach yields finely tuned parameters while ensuring computational efficiency remains satisfactory. In Table 1 and Table 2, we illustrate an example of one of the hyper-tuned CNNs. Notably, for the CNN with mean and standard deviation output, 14 out of the 21 hyper-tuned configurations converge to the same set of hyperparameters for the model.

Conv_layers	**3**
Conv_1 filters	71
Conv_1 activation	tanh
Conv_2 filters	101
Conv_2 activation	relu
Conv_3 filters	101
Conv_3 activation	relu

Table 6: CNN with dropout

Conv_layers	**3**
Conv_1 filters	121
Conv_1 activation	tanh
Conv_2 filters	11
Conv_2 activation	relu
Conv_3 filters	11
Conv_3 activation	tanh

Table 7: CNN with mean stddev

In the case of the CNN with MC dropout, the β value defaults to 2.6, which provides sufficient exploration across the search space. However, for the CNN with mean and standard deviation output, the β value settles at 18, indicating a necessity for increased exploration across the search space. Since the β value is not a hypertunable value this value will need to be determined by trial and error.

LSTM

In the LSTM hyperparameter tuning process with dropout, the tuning varies significantly across each fold, although all models ultimately converge to utilizing 3 LSTM layers. It's worth noting that only the ReLU and tanh activation functions are employed. Alternatively, in the LSTM with mean and standard deviation output, the sigmoid activation function is tuned. Furthermore, the number of layers differs among 3, 4, and 6 for the LSTM with mean and standard deviation output.

LSTM_layers	**3**
LSTM_1 units	176
LSTM_1 activation	relu
LSTM_2 units	166
LSTM_2 activation	tanh
LSTM_3 units	76
LSTM_3 activation	tanh

Table 8: LSTM with dropout

LSTM_layers	**3**
LSTM_1 units	18
LSTM_1 activation	sigmoid
LSTM_2 units	142
LSTM_2 activation	sigmoid
LSTM_3 units	14
LSTM_3 activation	relu

Table 9: LSTM with mean stddev

For the LSTM we see a same behaviour as with the CNN for the β value. The LSTM with MC dropout stayes on the default value of 2.6, whereas the value for the LSTM with mean stddev output is needed to increase to 12 to explore enough over the search space.

MDN and BNN

In the case of the MDN, the number of output distributions stands out as one of the most crucial hyperparameters. During hyperparameter tuning, the options typically include 2 or 5 distributions. For all but one model, hyperparameter tuning results in 6 dense layers. On the other hand, for the BNN, all models tune to the same set of hyperparameters. These results are shown in Table 11. For the BNN, the maximum number of units for hyperparameter tuning was set to 64, as increasing the number of units in each Bayesian layer significantly extends the tuning duration due to the computational complexity of the BNN compared to other algorithms. Interestingly, the BNN consistently yields the same hyperparameters across all 21 folds, indicating uniformity and robustness across different subsets of the data.

MDN_output_dimensions	**2**
Dense_layers	**6**
Dense_1 units	128
Dense_1 activation	relu
Dense_2 units	128
Dense_2 activation	tanh
Dense_3 units	8
Dense_3 activation	tanh
Dense_4 units	128
Dense_4 activation	relu
Dense_5 units	8
Dense_5 activation	tanh
Dense_6 units	8
Dense_6 activation	relu

Table 10: MDN

BNN_layers	**2**
BNN_1 units	6
BNN_1 activation	relu
BNN_1 kl_use_exact	True
BNN_1 kl weight	0.0001
BNN_2 units	24
BNN_2 activation	relu
BNN_2 kl_use_exact	False
BNN_2 kl weight	0.0001
dense neurons	198

Table 11: BNN

For both the MDN and BNN, the β value is determined to be 16. This value is chosen to ensure sufficient exploration across the search space during the hyperparameter tuning process.

Overall configuration

For all algorithms, we will employ a ReLU activation function with a maximum value of 100. This choice is based on the understanding that the SOH value cannot exceed 100, since the SOH can't be above the 100%.

6.2 Results SOH

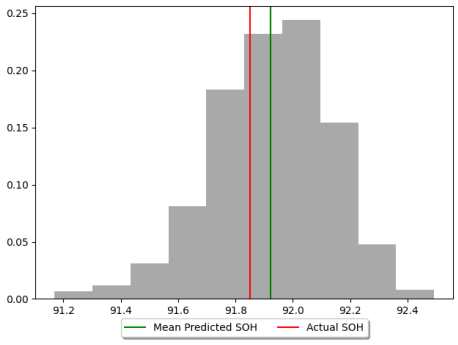
In this section, we present the comprehensive results for each of the algorithms. Table 12 displays the MAE and RMSE, providing an overview of the performance of each algorithm. From the results, it is evident that in terms of these generic metrics, the LSTM employing MC dropout performs the best, followed by the MDN. Alternatively, the BNN exhibits the poorest performance for predicting SOH.

Algorithms		MAE	RMSE
CNN	*dropout*	1.58	1.88
	mean	2.37	2.90
LSTM	*dropout*	1.29	1.62
	mean	1.86	2.18
MDN		1.39	1.63
BNN		4.56	5.45

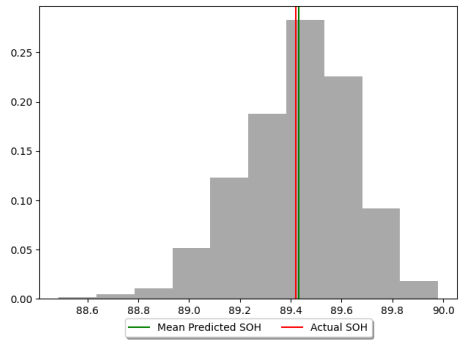
Table 12: SOH Overall results

In Appendix 10.3, the general metrics for each of the VAH profiles are presented, along with the overall general score. Initially, let's focus on the *baseline* mission profiles (VAH01, VAH17, VAH27). Based on the overall scores, the LSTM with MC dropout is the best performer. For VAH01, the LSTM with MC dropout performs better compared to the other algorithms, achieving an MAE score of 1.1 and an RMSE score of 1.3, compared to the second best performer, the MDN, with scores of 2.29 and 2.45, respectively. however, the BNN outperforms the CNN with mean and stddev output, scoring an MAE and RMSE of 3.48 and 4.22, respectively, compared to 4.04 and 5.59 for the CNN. Moving to VAH17, the LSTM with MC dropout again leads the pack with an MAE of 0.72 and an RMSE of 0.81, closely followed by the MDN with scores of 0.84 and 0.91, respectively. The CNN with mean and standard deviation output performs the poorest for VAH17, with scores of 5.34 and 7.17, while the BNN follows with scores of 4.42 and 5.51. For the last baseline profile, VAH27, we observe the MDN outperforming the LSTM with MC dropout, with scores of 0.62 and 0.71, respectively, compared to the LSTM's scores of 0.86 and 0.91. In Figure 9, we illustrate four resulting distributions for the LSTM with MC dropout. Figures 9a and 9b depict the 4th and 6th distributions for VAH01. The 4th input has a mean predicted value of 91.92 and a true value of 91.85, while the 6th input has a predicted mean of 89.43 and a true value of 89.42, indicating distributions that predict values close to the true SOH with differences of 0.07 and 0.01, respectively.

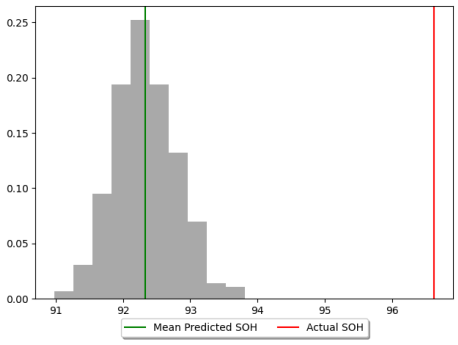
VAH07 represents a distinct mission profile utilized solely during SOH prediction due to its unique characteristic of not reaching the 85% capacity test limit. Upon examining the performance of various algorithms for VAH07, it becomes evident that all algorithms, except for the CNN with MC dropout, exhibit notably inferior performance compared to other profiles. Figures 9c and 9d illustrate the distributions of the initial and final inputs for VAH07, respectively. The initial distribution of VAH07 displays a difference of 4.29% between the predicted mean and the true value, while the final distribution exhibits a difference of 5.63%. This variance can be attributed to VAH07 having the fewest capacity tests among all profiles. Moreover, considering the significance of all features, it is apparent that VAH07's overall inferior performance may also be attributed to factors such as the duration of CC and CV time. Notably, Appendix 10.2 demonstrates that the feature related to CV duration holds high importance, with VAH07 distinguishing itself from baseline missions by maintaining a CV voltage of 4.0V, as opposed to the standard 4.2V. To validate these suspicions, we will examine VAH23, which is the only other mission profile featuring a variation in the CV voltage. Interestingly, even the most effective algorithm, namely the LSTM with MC dropout, achieves only modest scores of 2.91 and 3.26, respectively. Figure 10 illustrates the CC and CV values for the initial five capacity tests for VAH07 alongside those for each baseline mission profile. Here, it becomes evident that the CC and CV values of VAH07 are significantly lower compared to those of the baseline missions. Additionally, the figure depicts the CC and CV values for VAH23, the only other mission profile with variations in CV voltage, set at 4.1V in this instance. As demonstrated in Appendix 10.3, the performance of all algorithms for VAH23 is notably poorer. Particularly, when considering the LSTM with



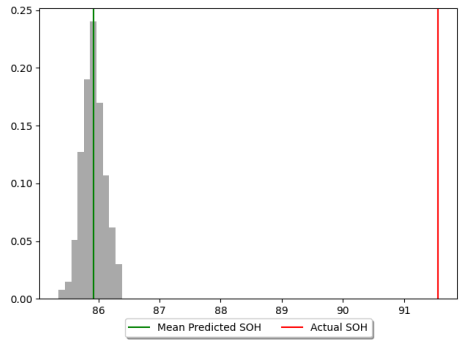
(a) VAH01 4th input



(b) VAH01 6th input



(c) VAH07 1st input



(d) VAH07 4th input

Figure 9: LSTM with MC dropout distributions VAH01 and VAH07

MC dropout, the MAE and RMSE are 2.91 and 3.26, respectively, rendering VAH23 the second poorest performing mission profile, following VAH07. This observation underscores the significant impact of CV voltage variations on the accuracy of SOH predictions.

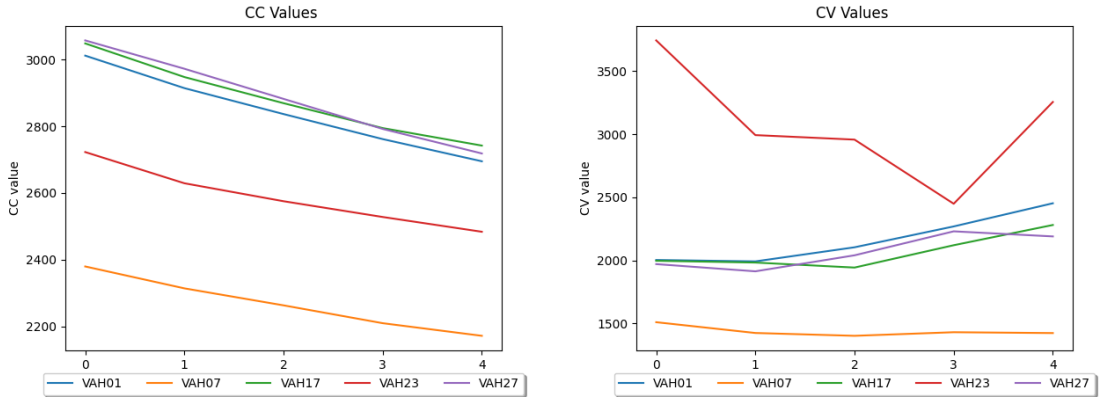


Figure 10: CC and CV values VAH07

Novel metrics

The Generic metrics offer a broad overview of algorithm performance for SOH predictions, relying solely on the mean values of each distribution. However, this approach overlooks the unique characteristics inherent in each distribution. In Appendix 10.4, a comprehensive summary of all novel metrics is presented. As elaborated in section 4.2.6, these novel metrics provide a more nuanced understanding of the intricacies within prognostic predicted distributions.

Algorithms		**CRPS**	**CRPS weighted**	**rs over**	**rs under**
CNN	*dropout*	1.443	1.555	0.000	0.368
	mean	2.780	2.760	0.005	0.043
LSTM	*dropout*	1.172	1.398	0.000	0.423
	mean	2.381	2.248	0.006	0.057
MDN		1.088	1.203	0.000	0.280
BNN		3.130	2.916	0.035	0.001

Table 13: SOH novel metric results

In Table 13, we present a comprehensive overview of the novel metrics for SOH predictions. Notably, the *CRPS* and *CRPS weighted* scores, alongside the *rs over* and *rs under* scores, offer valuable insights into the characteristics of the predicted distributions. Understanding these over and underestimations is crucial in PHM, especially in scenarios where overestimation can lead to significant consequences, such as early component replacement in EVTOL batteries to avoid in-flight failures. Analyzing the *CRPS* scores, which evaluate distribution accuracy and sharpness, we observe that the MDN outperforms the LSTM with

MC dropout, scoring 1.088 and 1.172, respectively. This suggests that the MDN exhibits better sharpness in its predictions compared to the LSTM with MC dropout. Considering the *rs over* and *rs under* scores, which offer insights into over and underestimations, we note that the CNN, LSTM with MC dropout, and MDN exhibit minimal to no overestimation, as indicated by their *rs over* score of 0.000. However, these algorithms display varying degrees of underestimation. Conversely, the CNN, LSTM with mean stddev output, and BNN demonstrate some overestimation. Particularly, the BNN shows an overall *rs over* score of 0.0356, indicating that predictions from the CNN, LSTM with MC dropout, and MDN are safer in critical component operations. Despite the conventional metrics highlighting VAH07's poorer performance due to data variations, this behavior is also reflected in the novel metrics, where the MDN still outperforms VAH07 with a CRPS score of 8.178. Overall, based on the novel metrics, both the LSTM with MC dropout and MDN exhibit commendable performance with decent CRPS scores and no overestimation

7 Estimating the distribution of RUL (probabilistic prognostics)

In the following section, we will delve into the results concerning the prognostic predicted distributions of the RUL. Similar to the preceding section, we will utilize the 24 most crucial features identified through computed SHAP values, as outlined in section 4.2.1 for RUL predictions. Subsequently, we will discuss the outcomes of hyperparameter tuning for each algorithm, followed by the evaluation results employing both general metrics and novel metrics. It's important to note that for RUL predictions, VAH07 cannot be utilized due to its failure to reach the EOL threshold of 85%.

7.1 Hypertuning results RUL

For hyperparameter tuning, a similar approach as with the SOH predictions will be employed. For each algorithm we will be utilizing 150 trials and 60 epochs in each trial. Firstly, we will examine the hypertuned parameters of the CNNs, followed by the LSTMs, and finally, the MDN and BNN. In Table 14, the hypertuned parameters for the CNNs are presented. Notably, there are 3 convolutional layers, contrasting with Table 7, which represents 4 convolutional layers. For both the CNN with MC dropout and CNN with mean stddev output, the default parameter α is utilized. The default value of β is 2.6 for the CNN with Monte Carlo dropout, while for the CNN with mean stddev output, it requires a significantly increased value of 18. As discussed in section 4.2.5, β indicates the factor determining the exploratory nature of the Bayesian Optimization tuner within the search space.

Conv_layers	**3**
Conv_1 filters	211
Conv_1 activation	tanh
Conv_2 filters	91
Conv_2 activation	tanh
Conv_3 filters	171
Conv_3 activation	relu

Table 14: CNN with dropout

Conv_layers	**3**
Conv_1 filters	21
Conv_1 activation	tanh
Conv_2 filters	111
Conv_2 activation	relu
Conv_3 filters	21
Conv_3 activation	tanh
Conv_4 filters	191
Conv_4 activation	tanh

Table 15: CNN with mean stddev

LSTM

Following the analysis of CNNs, let's delve into the hyperparameters for both LSTM algorithms. For the LSTM with MC dropout, the hypertuning process concludes with 3 layers, while for the LSTM with mean stddev output, it settles on 6 LSTM layers. For both algorithms, default values are employed for the α and β parameters in the case of LSTM with MC dropout. However, for the LSTM with mean stddev output, the β value is utilized. Notably, the hypertuning process determines a β value of 12, indicating that similar to the CNN with mean stddev output, there is a need for increased exploration to optimize the hyperparameters.

LSTM_layers	**3**
LSTM_1 units	256
LSTM_1 activation	relu
LSTM_2 units	196
LSTM_2 activation	tanh
LSTM_3 units	196
LSTM_3 activation	tanh

Table 16: LSTM with dropout

LSTM_layers	**6**
LSTM_1 units	202
LSTM_1 activation	tanh
LSTM_2 units	178
LSTM_2 activation	relu
LSTM_3 units	238
LSTM_3 activation	relu
LSTM_4 units	214
LSTM_4 activation	tanh
LSTM_5 units	98
LSTM_5 activation	relu
LSTM_6 units	190
LSTM_6 activation	relu

Table 17: LSTM with mean stddev

MDN and BNN

Unlike the previous algorithms which converged to a majority choice of hyperparameters, the MDN exhibits a 50% split, resulting in either a 3-layer or 4-layer MDN, as shown in Table 18. On the other hand, all folds for the BNN converge to a 2-layered architecture, with 6 and 24 units respectively, culminating in a dense layer comprising 198 units. Similar to the CNN and LSTM with mean and stddev output, both the MDN and BNN, which output multiple and a single normal distribution respectively, necessitate an increased β value of 16 to ensure adequate exploration of the search space.

MDN_output_dimensions	**3**	**3**
Dense_layers	**3**	**4**
Dense_1 units	92	60
Dense_1 activation	tanh	tanh
Dense_2 units	82	106
Dense_2 activation	relu	relu
Dense_3 units	104	124
Dense_3 activation	relu	relu
Dense_4 units	***	12
Dense_4 activation	***	relu

Table 18: MDN

BNN_layers	**2**
BNN_1 units	6
BNN_1 activation	relu
BNN_1 kl_use_exact	True
BNN_1 kl weight	0.0001
BNN_2 units	24
BNN_2 activation	relu
BNN_2 kl_use_exact	False
BNN_2 kl weight	0.0001
dense neurons	198

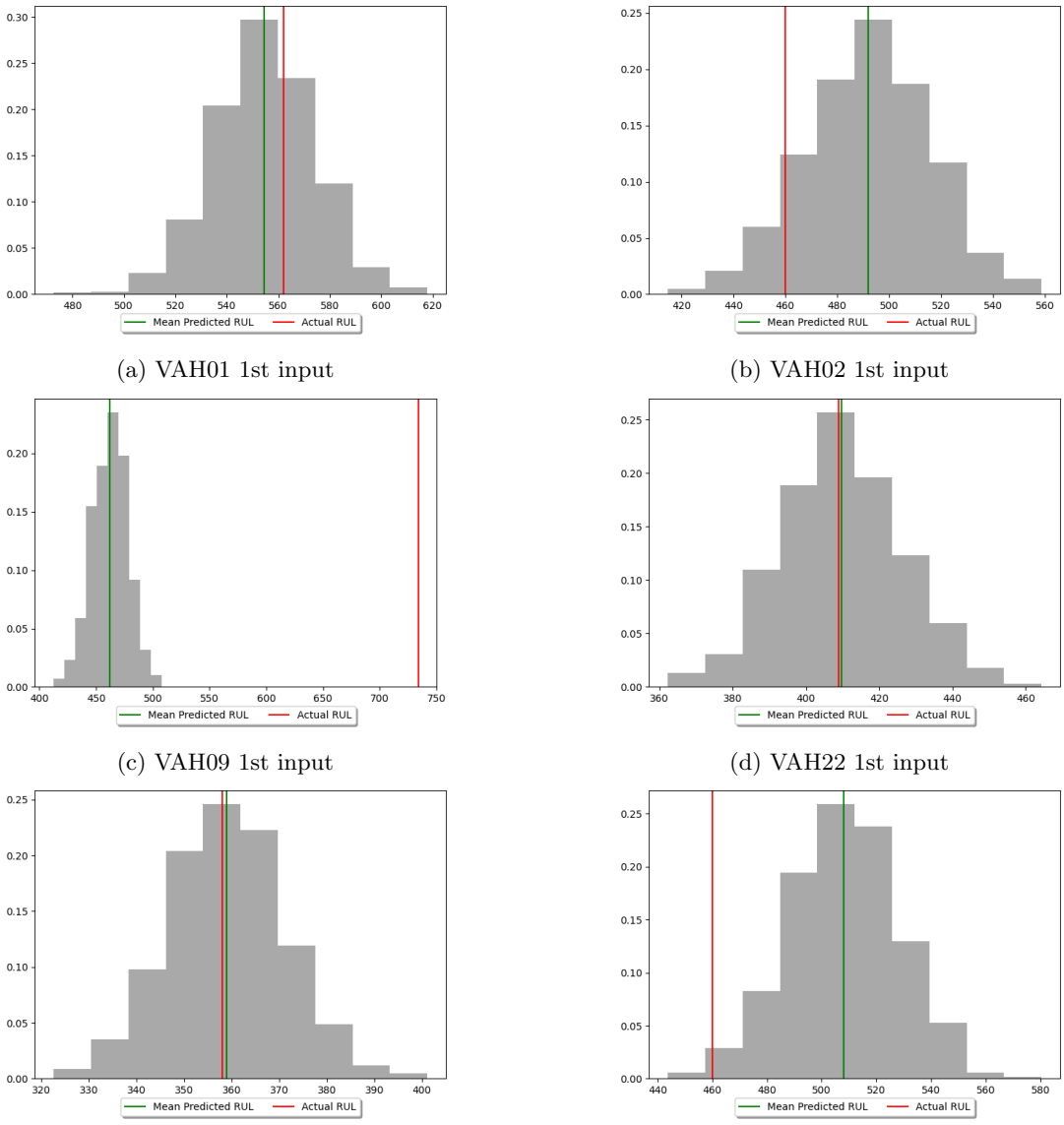
Table 19: BNN

7.2 Results RUL

In this section, we discuss the results obtained for each algorithm concerning the RUL predictions. Similar to the SOH predictions, we compare the outcomes of generic metrics with novel metrics. Table 20 presents the generic results for probabilistic distribution predictions, revealing that for both MAE and RMSE scores, the MDN achieves the best performance, with a MAE of 66.66 flights and an RMSE of 76.25 flights, respectively.

Algorithms		**MAE**	**RMSE**
CNN	*dropout*	70.10	81.25
	mean	79.36	92.39
LSTM	*dropout*	81.23	91.17
	mean	69.11	79.74
MDN		66.66	76.25
BNN		161.72	188.62

Table 20: RUL Overall results



(a) VAH01 1st input

(b) VAH02 1st input

(c) VAH09 1st input

(d) VAH22 1st input

(e) VAH22 2nd input

(f) VAH30 1st input

Figure 11: MDN for VAH01, VAH02, VAH09, VAH22 and VAH30

In Appendix 10.5, the generic metrics are provided for each algorithm across various mission profiles. Firstly, focusing on the baseline mission profiles, we observe that for VAH01, the CNN with mean stddev output achieves the best performance with a MAE of 69.74. Conversely, for VAH17, the LSTM with mean stddev output emerges as the top-performing algorithm, followed closely by the MDN. Similarly, for VAH27, the LSTM with mean stddev output outperforms other algorithms, with the LSTM with MC dropout as the

runner-up. Unlike in the SOH predictions where VAH07 consistently underperformed due to variations in CV voltage, in the RUL predictions, different mission profiles exhibit varying levels of performance disparities. As discussed in section 4.2.1, the maximum temperature during landing emerges as a crucial factor. Notably, VAH09 (20°C), VAH25 (20°C), and VAH30 (35°C) perform relatively poorly compared to other mission profiles. Variations in ambient temperature, typically set at 25 degrees, significantly impact the maximum temperature during landing. For the MDN, which achieves the best overall performance with a MAE of 66.66, the MAE for VAH09, VAH25, and VAH30 are notably higher at 127.15, 110.22, and 117.14, respectively. alternatively, for the second-best performing algorithm, LSTM with mean stddev output, the MAE scores for these three mission profiles are 137.36, 141.19, and 105.27, resulting in an average MAE of 69.11. In Figure 12, the plotted maximum temperatures for each phase are shown for the first 10 capacity tests. It is evident that the temperature trends for the baseline mission profiles remain relatively close together, whereas VAH30, with a temperature increase of 10° Celsius, exhibits significantly higher maximum temperatures for each phase during flight. Similarly, VAH09 and VAH25 display lower maximum temperatures compared to the baseline missions. This discrepancy in temperatures compared to the baseline mission profiles highlights the effectiveness of maximum temperature as a representation of battery age. An intriguing observation is the low MAE of VAH22, particularly for the LSTM with Monte Carlo dropout, which achieves MAE scores of 8.83 and 7.21 compared to the averages of 66.66 and 81.23 for these algorithms, respectively. However, it is notable that the performance of RUL predictions is less Explicable compared to SOH predictions. In Figure 11, four first input distributions are displayed for VAH01, VAH02, VAH09, VAH22, and VAH30. VAH01 and VAH02 exhibit close predictions to the true value, with the true value falling within the distribution range. Conversely, VAH09 and VAH30, which generally perform worse overall, display notable discrepancies. Specifically, VAH09, with a lower ambient temperature of 20°C, demonstrates significant overestimation in the first input distribution. Similarly, for VAH30, where the first input is underestimated. In contrast, VAH22, known for its accurate overall predictions, displays a prognostic predicted mean that slightly overestimates the true value.

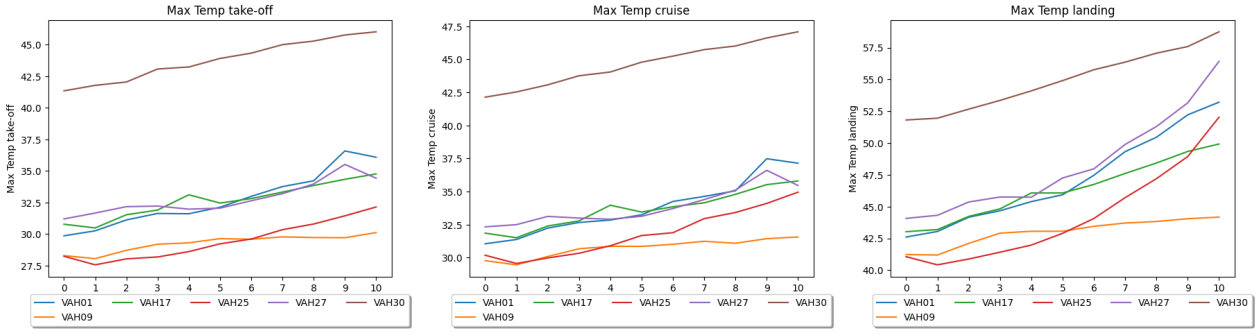


Figure 12: Max Temp for each flight phase

Novel metrics

Algorithms		CRPS	CRPS weighted	rs over	rs under
CNN	*dropout*	64.093	59.434	0.000	0.345
	mean	62.402	53.761	0.000	0.049
LSTM	*dropout*	80.503	87.999	0.000	0.453
	mean	62.108	51.670	0.000	0.171
MDN		62.881	57.323	0.000	0.382
BNN		114.403	107.408	0.002	0.045

Table 21: RUL novel metric results

As observed in the results for SOH, the generic metrics offer a general understanding of the distributions but fail to capture their characteristics fully. For RUL, the CRPS scores of CNN and LSTM with mean stddev output are both close, with scores of 62.402 and 62.108, respectively. Despite performing well in other generic metrics, the MDN slightly underperforms in CRPS with a score of 62.881. It's important to note that RUL measurements are taken every 51 missions, and the difference in CRPS scores between these algorithms is only 0.773 flights, indicating overall good performance. In terms of overall CRPS scores, the LSTM with MC dropout performs the worst, followed by the BNN. As with SOH predictions, knowing whether an algorithm will over or under predict is crucial. In Appendix 10.6, the Novel metrics are displayed. Both CNN and LSTM with MC dropout exhibit an overall underestimation of 0.5 rs under with no overestimation. The mean stddev output variants display similar trends, with underestimations of 0.452 and 0.470, respectively. The MDN shows an underestimation of 0.382, indicating it is not the smallest in underestimating overall. Conversely, the BNN displays even lower underestimation at 0.045. However, despite this, its higher CRPS score suggests lower accuracy compared to other algorithms. The increase in underestimation can potentially lead to earlier replacements when using the MDN. However, this would need to be balanced against the importance of early replacements for accident prevention.

8 Estimating the distribution of the RUC (probabilistic prognostics)

The RUC (Remaining Usefull Capacity tests) represents a reinterpretation of the RUL. While RUL denotes the number of remaining flights before reaching EOL, RUC signifies the remaining number of capacity tests before reaching EOL. In the first part, we will explore the hyperparameter tuning process for each algorithm. Subsequently, we will analyze the results using both generic and novel metrics. Similar to the RUL results, VAH07 will be excluded from the analysis since this mission profile does not reach the EOL threshold of 85

8.1 Hypertuning results RUC

For the hypertuning of the RUC, we will adopt the same approach as with the SOH and RUL predictions. Firstly, we will address the CNNs, followed by the LSTMs, and finally, the MDN and BNN. As observed in the SOH and RUL predictions, the β values remain consistent for both types of predictions. Therefore, for the RUC predictions, the β values will also remain the same as those used for SOH and RUL predictions.

Conv_layers	**3**
Conv_1 filters	121
Conv_1 activation	relu
Conv_2 filters	121
Conv_2 activation	relu
Conv_3 filters	131
Conv_3 activation	tanh

Table 22: CNN with dropout

Conv_layers	**4**
Conv_1 filters	71
Conv_1 activation	tanh
Conv_2 filters	171
Conv_2 activation	relu
Conv_3 filters	101
Conv_3 activation	tanh
Conv_4 filters	181
Conv_4 activation	tanh

Table 23: CNN with mean stddev

For the CNN with Monte Carlo dropout, we observe a triple-layer architecture consisting of 121, 121, and 131 units respectively, with ReLU, ReLU, and tanh activation functions, as illustrated in Table 22. On the other hand, for the CNN with mean and stddev output, a four-layer architecture is utilized with 71, 171, 101, and 181 units respectively, along with tanh, ReLU, tanh, and tanh activation functions, as displayed in Table 23.

LSTM

LSTM_layers	**3**
LSTM_1 units	146
LSTM_1 activation	relu
LSTM_2 units	176
LSTM_2 activation	relu
LSTM_3 units	126
LSTM_3 activation	tanh

Table 24: LSTM with dropout

LSTM_layers	**2**
LSTM_1 units	242
LSTM_1 activation	tanh
LSTM_2 units	46
LSTM_2 activation	relu

Table 25: LSTM with mean stddev

For the LSTM with MC dropout the configuration of three layers with 146, 176 and 126 is tuned with ReLU, ReLU and tanh used in order as we can see in table24. For this algorithm the β value is settled on the default value of 2.6. For the LSTM with mean stddev output the hypertuning is settled on a smaller model of only 2 LSTM layers. The first layer is having 242 units and is using the tanh activation function, while the second layer contains 46 units and uses the ReLU function.

MDN and BNN

MDN_output_dimensions	**3**	**5**
Dense_layers	**4**	**3**
Dense_1 units	60	68
Dense_1 activation	tanh	tanh
Dense_2 units	106	58
Dense_2 activation	relu	tanh
Dense_3 units	124	16
Dense_3 activation	relu	relu
Dense_4 units	12	***
Dense_4 activation	relu	***

Table 26: MDN

BNN_layers	**2**
BNN_1 units	12
BNN_1 activation	tanh
BNN_1 kl_use_exact	True
BNN_1 kl weight	0.00001
BNN_2 units	24
BNN_2 activation	sigmoid
BNN_2 kl_use_exact	False
BNN_2 kl weight	0.0001
dense neurons	246

Table 27: BNN:

For the MDN there are two majority configuration during the hypertune which are shown in 26. First configuration of hyperparameters is with three output distributions and 4 dense layers. The second configuration contains five output distributions and containing 3 dense layers. The first layer is used for 9/19 folds where are the second configuration is used for 10/19 folds. For the MDN the β value that is used was settled on 16, this gave better exploration over the search space for optimization. For the BNN the hypertuning is settled to a single configuration of two BNN layers, which is shown in table 27 the first layer contains 12 units with tanh and the second layer contains 24 units with sigmoid as activation function.

8.2 results RUC

In this section, we will discuss the results for the prognostic predicted distributions of the RUC. Firstly, we will examine the overall scores using the generic metrics, as shown in Table 28. Based on the MAE and RMSE, the LSTM with MC dropout emerges as the best-performing algorithm with scores of 1.31 and 1.52 respectively. The CNN with MC dropout closely follows as a close second with a MAE score of 1.33, representing a minor difference of only 0.02. In contrast, the BNN performs the worst overall with MAE and RMSE scores of 2.08 and 2.48 respectively.

Algorithms		**MAE**	**RMSE**
CNN	*dropout*	1.33	1.53
	mean	1.60	1.94
LSTM	*dropout*	1.31	1.52
	mean	1.42	1.60
MDN		1.44	1.64
BNN		2.08	2.48

Table 28: RUC Overall results

In Appendix 10.7, an overview of each mission profile for the generic metrics is provided. Notably, VAH09 consistently emerges as one of the worst-performing mission profiles across various algorithms. This may potentially correlate with the feature importance of temperature, as the maximum temperature during

take-off, cruise, and landing holds significant importance. For instance, VAH09 yields a MAE of 2.81 and RMSE of 3.40 for the LSTM with MC dropout, making it the worst-performing mission profile for this algorithm. A closer examination reveals VAH25 and VAH30 as the second and third worst-performing profiles, respectively. These profiles exhibit alterations in ambient temperature, as illustrated in Figure 12. For the second-best algorithm, CNN with MC dropout, VAH09, VAH25, and VAH30 yield MAE scores of 2.70, 2.25, and 2.67, respectively. The differences in maximum temperature compared to baseline mission profiles are evident. Analyzing the feature importance of the RUC, besides maximum temperatures, voltage values during take-off, as well as CC and CV timings, also hold significance. Since differences in CV voltage can significantly impact voltage values during each flight phase, mission profiles like VAH07, which deviates from standard CV voltage, are disregarded from RUC predictions. Similar effects of CV and CC values on SOH predictions were observed in section 6. For example, VAH23, with a CV voltage of 4.1V compared to the standard 4.2V, exhibits a MAE of 1.85 and RMSE of 2.22 for LSTM with MC dropout, slightly higher than the overall scores. VAH24, with an alteration in C-rate of 0.5C, ranks as the fourth worst-performing mission profile with a MAE of 1.89 and RMSE of 1.94.

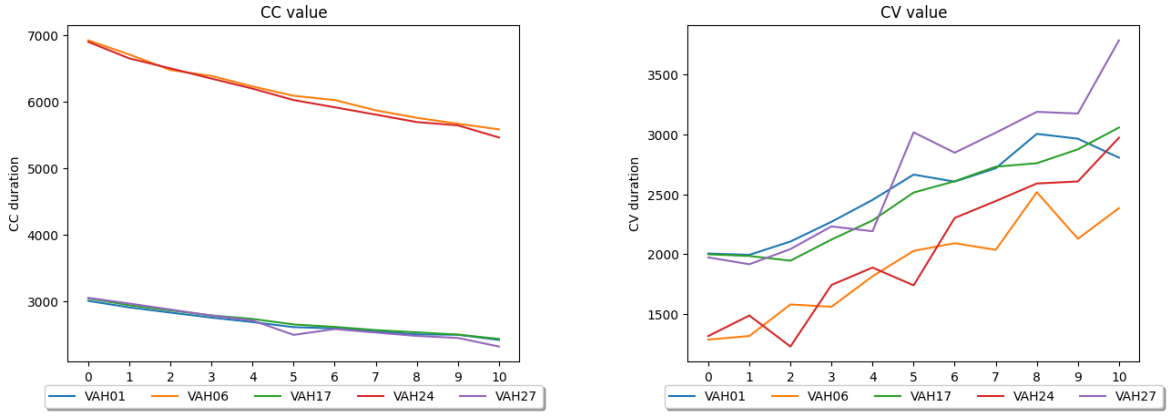


Figure 13: CC and CV values VAH06 and VAH24

In Figure 13, a comparison between VAH24 and the baseline mission profiles is described, highlighting significant differences, especially in the CC value. The duration of the CC charging phase increases notably for VAH24 due to its CC-rate of 0.5C, which substantially delays the CC-charging phase. For instance, in VAH01 (a baseline mission profile), the first capacity test has a CC duration of 3012, whereas in VAH24, the first capacity test has a CC duration of 6893, representing an increase of 2.28 times. Interestingly, the plotted values also include the CC and CV duration's of VAH06, which closely align with those of VAH24. This observation is noteworthy because VAH06 also has a C-rate of 0.5C, indicating the influence of this alteration on the charging profile.

Novel metrics

In this section, we present the novel metrics for the RUC predictions. Table 29 provides an overview of the overall novel metrics. It is observed that the LSTM with mean stddev output performs the best in terms of CRPS, achieving a score of 1.180. Following closely are the CNN and LSTM with MC dropout, ranking

second and third in CRPS scores. These algorithms, utilizing MC dropout, achieve CRPS scores of 1.243 and 1.284 respectively.

Algorithms		**CRPS**	**CRPS weighted**	**rs over**	**rs under**
CNN	*dropout*	1.243	1.175	0.000	0.378
	mean	1.284	1.314	0.000	0.153
LSTM	*dropout*	1.319	1.298	0.000	0.458
	mean	1.180	1.193	0.000	0.257
MDN		1.377	1.271	0.000	0.378
BNN		1.586	1.653	0.034	0.002

Table 29: RUC novel metric results

For the CRPS weighted score, the ranking of algorithms differs compared to the CRPS, where the LSTM with MC dropout, previously the best-performing, is outperformed by the CNN with MC dropout. The difference between these two algorithms is 0.018. This difference suggests that the CNN with MC dropout tends to have a higher underestimation compared to the LSTM with mean stddev output. This observation is supported by the RS under, which is higher for the CNN with MC dropout (0.378) compared to the LSTM with mean stddev output (0.257). In contrast, the worst-performing algorithm is the BNN, with a CRPS of 1.586 and an RS under of 0.002. Notably, the BNN is the only algorithm with an rs over score of 0.034, indicating some level of overestimation.

8.3 RUC as Conversion of the RUL

In this research, we have introduced the concept of RUC, which serves as a representation of RUL. In this section, we will recalculate the RUL from the RUC data to evaluate the effectiveness of translating labels between these two representations. Recalculating the RUL from the RUC data involves deriving estimates of the remaining lifespan of the batteries based on the available capacity test data. This process allows us to assess whether the RUC data accurately reflects the remaining useful life of the batteries and whether it can be effectively translated into RUL estimates. By comparing the recalculated RUL values with the original RUL data, we can determine the accuracy and reliability of the RUC representation. If the recalculated RUL values closely align with the original RUL data, it would indicate that the RUC representation effectively captures the RUL of the batteries. This analysis will provide valuable insights into the utility of RUC data and its potential to serve as a viable alternative or complementary representation to traditional RUL estimates.

Algorithms		**MAE**			**CRPS**		
		RUL	RUC	(RUC x51)	RUL	RUC	(RUC x51)
CNN	dropout	70.10	1.33	67.83	64.093	1.243	63.393
	mean	79.36	1.60	81.60	62.402	1.284	65.484
LSTM	dropout	81.23	1.31	66.81	80.503	1.319	67.269
	mean	69.11	1.42	72.42	62.108	1.180	60.180
MDN		66.66	1.44	73.44	62.881	1.377	70.227
BNN		161.72	2.08	106.08	114.403	1.586	80.886

Table 30: Results RUC and RUL

In Table 30, the results of both RUL and RUC are presented, along with the recalculated RUL derived from the RUC data. The recalculated RUL values are obtained by multiplying the RUC values by 51, as each RUC measurement is taken every 51 missions. Upon examination, it is observed that for the best performing algorithm in terms of RUL MAE, the MDN with a MAE of 66.66, the recalculated RUL from the RUC is 73.44. This suggests that the RUC representation may not fully capture the remaining useful life of the batteries accurately, as the recalculated RUL values differ from the original RUL data. Interestingly, the only algorithms that seem to benefit from the recalculated RUL from RUC are the CNN and LSTM with MC dropout, along with the BNN. This implies that for these algorithms, the RUC data may provide useful information that can be effectively translated into RUL estimates. Overall, these findings highlight the importance of carefully evaluating the translation between RUC and RUL representations and suggest that further investigation is needed to fully understand the effectiveness of using RUC data for predicting remaining useful life.

9 Conclusion

In the foreseeable future, eVTOL aircraft hold significant potential for addressing growing urban mobility needs and alleviating traffic congestion in densely populated areas. Leveraging Lithium-ion batteries for their notable energy density and cost-effectiveness, eVTOLs are poised to play a pivotal role in urban transportation. However, effective battery management systems are critical to ensuring the reliability and safety of eVTOL operations, particularly to mitigate potential failures during flight. In this thesis we have explored the utilization of different algorithms for prognostic distribution for the state of health of batteries of the Vahana Evtols. The importance of understanding under and overestimation of the prediction within PHM systems is of high importance, for scheduling maintenance of the batteries. The dataset that is utilized is a combination of the Vahana dataset with added the impedance data that is presented on 21 April 2023. the dataset consists of timestamps with different values of the Vahana battery, such as cell voltage(V), amount of energy supplied(mAh) and amount of voltage discharged(mAh). We have explored the use of different algorithms for the prognostic distribution of these batteries, such as the use of CNN and LSTM using MC dropout. Following we also utilized a CNN and LSTM using a output of a normal distribution by the mean and stddev values. In addition a BNN which utilizes prior and posterior distributions as weights and outputs a normal distribution and a MDN which outputs multiple distributions are tested. Three different values are tested for each of the algorithms, first we used the SOH(State Of Health) which is the remaining max capacity the battery can still hold in percentage, following by the RUL(Remaining Usefull Life) which is the remaining amount of flights the battery can still be used before hitting the EOL of 85%. The last value for which distributions was predicted is the RUC(Remaining Useful Capacity tests), which is a representation of the RUL in terms of remaining capacity tests left for each of the mission profiles before hitting the EOL. For the validation of algorithms we have utilized two sets of metrics: generic metrics(MAE and RMSE) and novel metrics(CRPS, RS over, RS under).

- For the SOH the best performing algorithm is the LSTM with MC dropout, with the MDN slightly outperforming the LSTM with MC dropout for the novel metrics.
- For the RUL the MDN is best performing for the generic metrics and the LSTM with mean stddev output is best performing with the novel metrics.
- For the RUC the LSTM with MC dropout is best performing for the generic metrics and for the novel metrics LSTM with mean stddev output is best performing

Future work

Building on the observations and results garnered in this study, the use of LSTM based algorithms could be futher explored for prediction of prognostic distributions. Following the characteristics of LSTM based models, which work well on a wide range of sequential based data. An other observation that has been seen in this study is the well performing implementations of MDNs which allows for mixtured distributions, by combining multiple probability distributions. In future study the exploration of utilizing implementations using combined techniques could have the potential of unlocking new ways for prediction of prognostic distributions. Combining the benefits of LSTM layers, which have major advantages on sequential data and the well performing MDNs on the Vahana dataset.

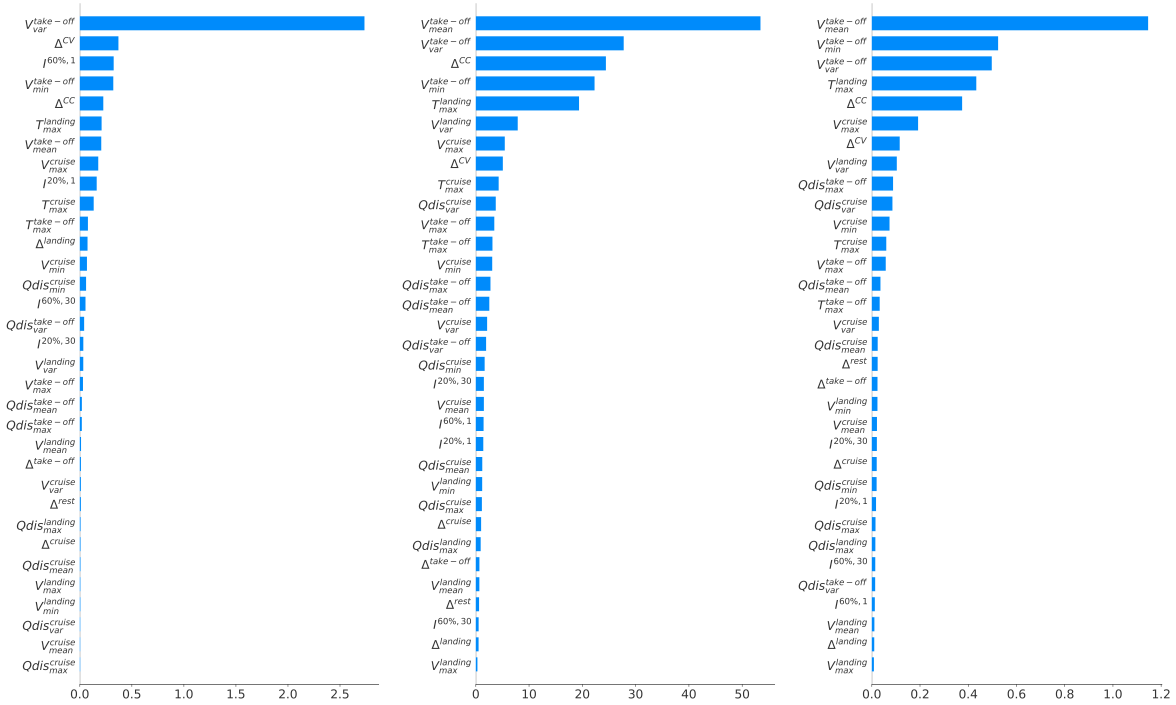
10 Appendix A

10.1 Appendix A-1: Overview of mission profile specifications

File	Cruise Duration	power usage Take-off (Watt)	power usage Cruise (Watt)	power usage Landing (Watt)	CC rate	CV	Ambient temp	missions
VAH01	800	54W	16W	54W	1-C	4.2V	$25°C$	847
VAH02	***1000***s	54W	16W	54W	1-C	4.2V	$25°C$	625
VAH05	800s	***48.6W***	***14.4W***	***48.6W***	1-C	4.2V	$25°C$	1615
VAH06	800s	54W	16W	54W	**0.5-C**	4.2V	$25°C$	9290
VAH07	800s	54W	16W	54W	1-C	**4.0V**	$25°C$	339
VAH09	800s	54W	16W	54W	1-C	4.2V	***20***$°C$	8527
VAH10	800s	54W	16W	54W	1-C	4.2V	***30***$°C$	1431
VAH11	800s	***43.2W***	***12.8W***	***43.2W***	1-C	4.2V	$25°C$	2249
VAH12	***400***s	54W	16W	54W	1-C	4.2V	$25°C$	2349
VAH13	***600***s	54W	16W	54W	1-C	4.2V	$25°C$	1042
VAH 15	**1000s**	54W	16W	54W	1-C	4.2V	$25°C$	554
VAH16	800s	54W	16W	54W	**1.5-C**	4.2V	$25°C$	559
VAH17	800s	54W	16W	54W	1-C	4.2V	$25°C$	1002
VAH20	800s	54W	16W	54W	**1.5-C**	4.2V	$25°C$	611
VAH22	***1000***s	54W	16W	54W	1-C	4.2V	$25°C$	579
VAH23	800s	54W	16W	54W	1-C	**4.1V**	$25°C$	697
VAH24	800s	54W	16W	54W	**0.5-C**	4.2V	$25°C$	801
VAH25	800s	54W	16W	54W	1-C	4.2V	***20***$°C$	554
‘ VAH26	***600***s	54W	16W	54W	1-C	4.2V	$25°C$	1164
VAH27	800s	54W	16W	54W	1-C	4.2V	$25°C$	587
VAH28	800s	***48.6W***	***14.4W***	***48.6W***	1-C	4.2V	$25°C$	1182
VAH30	800s	54W	16W	54W	1-C	4.2V	***35***$°C$	919

Table 31: Alternations in VAH mission profiles

10.2 Appendix A-2: feature importance based on the shap values



(a) Shap values for the SOH (b) Shap values for the RUL (c) Shap values for the RUC

Figure 14: Feature importance obtained with Shap values.

10.3 Appendix A-3: SOH results generic metrics

	VAH01	VAH02	VAH05	VAH06	VAH07	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	VAH_ALL
mae	2.35	1.41	1.41	1.15	1.63	1.21	1.68	2.51	1.49	1.42	0.64	1.11	0.89	1.56	3.78	0.88	1.70	0.69	2.43	1.68	1.58
rmse	2.49	1.69	1.59	1.50	2.06	1.47	2.21	3.36	1.88	1.49	0.73	1.35	1.05	1.92	4.26	0.95	1.74	0.91	2.83	2.10	1.88

Figure 15: SOH results MAE RMSE CNN

	VAH01	VAH02	VAH05	VAH06	VAH07	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	VAH_ALL
mae	1.1	0.46	1.08	1.48	5.40	1.42	1.08	2.75	0.71	0.41	0.61	0.72	0.96	1.06	2.91	0.51	0.79	0.86	0.81	0.69	1.29
rmse	1.3	0.51	1.28	2.78	5.45	2.97	1.32	3.24	0.87	0.49	0.72	0.81	1.01	2.16	3.26	0.59	0.85	0.91	1.04	0.83	1.62

Figure 16: SOH results MAE RMSE LSTM

	VAH01	VAH02	VAH05	VAH06	VAH07	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	VAH_ALL
mae	4.04	0.73	1.34	3.21	6.42	1.65	1.56	2.65	1.86	2.09	0.85	5.34	1.77	0.51	3.97	1.87	1.20	1.08	1.06	4.13	2.37
rmse	5.59	0.96	1.52	3.85	6.69	2.11	1.73	3.15	2.26	2.22	1.16	7.17	2.07	0.60	5.46	2.37	1.47	1.30	1.32	4.96	2.90

Figure 17: SOH results MAE RMSE CNN mean stddev

	VAH01	VAH02	VAH05	VAH06	VAH07	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	VAH_ALL
mae	2.36	3.96	0.61	2.02	9.23	2.35	0.64	0.52	1.59	0.74	1.16	2.36	0.59	0.58	5.03	0.45	0.61	0.63	0.96	0.74	1.86
rmse	2.45	4.91	0.74	2.74	9.28	3.46	0.76	0.64	2.14	0.86	1.38	2.91	0.77	0.67	5.89	0.55	0.71	0.76	1.20	0.86	2.18

Figure 18: SOH results MAE RMSE LSTM mean stddev

	VAH01	VAH02	VAH05	VAH06	VAH07	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	VAH_ALL
mae	2.29	0.37	1.14	0.68	8.39	1.37	0.83	2.21	1.60	0.42	0.31	0.84	0.27	0.44	2.93	0.40	0.71	0.62	1.01	0.99	1.39
rmse	2.45	0.46	1.64	1.07	8.56	2.27	1.01	2.68	1.79	0.47	0.41	0.91	0.35	0.76	3.40	0.46	0.88	0.71	1.22	1.11	1.63

Figure 19: SOH results MAE RMSE MDN

	VAH01	VAH02	VAH05	VAH06	VAH07	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	VAH_ALL
mae	3.48	5.77	4.74	4.30	7.12	4.35	5.20	4.40	4.85	4.53	3.85	4.42	4.48	4.34	3.50	3.90	5.04	4.10	4.50	4.27	4.56
rmse	4.22	6.62	5.67	5.14	7.36	5.11	6.17	4.83	5.96	5.72	4.87	5.51	5.37	5.40	4.66	4.83	5.91	5.26	5.46	4.99	5.45

Figure 20: SOH results MAE RMSE BNN

10.4 Appendix A-4: SOH results Novel metrics

	VAH01	VAH02	VAH05	VAH06	VAH07	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	vah_total
crps	2.095	1.180	1.167	0.894	1.292	0.963	1.494	2.236	1.305	1.138	0.469	0.929	0.760	1.373	3.293	0.712	1.490	0.502	2.155	1.440	1.443
crps_W	1.048	1.690	1.737	0.820	0.813	0.503	2.191	3.185	1.942	1.708	0.247	1.361	0.433	0.686	4.452	0.359	0.745	0.601	1.080	0.908	1.555
cov - 0.5	0.000	0.000	0.138	0.105	0.250	0.091	0.111	0.111	0.105	0.000	0.200	0.158	0.273	0.000	0.000	0.067	0.000	0.273	0.091	0.176	0.106
mean width - 0.5	0.615	0.583	0.663	0.632	1.349	0.789	0.505	0.707	0.537	0.570	0.557	0.484	0.438	0.461	1.329	0.480	0.488	0.886	0.718	0.697	0.654
cov - 0.95	0.000	0.500	0.276	0.421	0.500	0.455	0.370	0.289	0.316	0.200	0.500	0.368	0.364	0.000	0.231	0.267	0.000	0.909	0.182	0.471	0.323
mean width - 0.95	1.791	1.775	2.004	1.870	3.749	2.443	1.536	2.129	1.562	1.753	1.669	1.446	1.269	1.325	3.431	1.462	1.468	2.848	2.134	2.085	1.945
rs over	0.000	0.000	0.000	0.000	0.008	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.002	0.000	0.006	0.000	0.000	0.000
rs under	0.499	0.396	0.359	0.355	0.191	0.363	0.370	0.363	0.350	0.471	0.272	0.306	0.314	0.500	0.435	0.392	0.499	0.096	0.416	0.323	0.368
rs total	0.499	0.396	0.359	0.355	0.199	0.363	0.370	0.363	0.350	0.471	0.272	0.306	0.314	0.500	0.435	0.393	0.499	0.102	0.416	0.323	0.368

Figure 21: SOH results novel metrics CNN

	VAH01	VAH02	VAH05	VAH06	VAH07	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	vah_total
crps	1.015	0.387	0.972	1.321	5.247	1.106	0.974	2.622	0.638	0.338	0.538	0.640	0.862	0.942	2.736	0.429	0.700	0.778	0.720	0.625	1.172
crps_W	0.642	0.531	0.806	1.946	2.623	0.588	1.346	3.820	0.828	0.177	0.797	0.959	1.293	0.792	4.104	0.232	0.512	0.389	1.067	0.743	1.398
cov - 0.5	0.125	0.083	0.000	0.053	0.000	0.091	0.037	0.067	0.000	0.100	0.000	0.053	0.000	0.200	0.000	0.067	0.000	0.000	0.182	0.235	0.067
mean width - 0.5	0.224	0.186	0.238	0.395	0.360	0.707	0.246	0.350	0.180	0.202	0.171	0.210	0.217	0.383	0.430	0.223	0.211	0.191	0.231	0.220	0.290
cov - 0.95	0.188	0.417	0.069	0.158	0.000	0.182	0.111	0.156	0.211	0.300	0.200	0.263	0.000	0.300	0.077	0.467	0.100	0.000	0.318	0.294	0.191
mean width - 0.95	0.664	0.643	0.718	1.053	1.077	2.496	0.728	1.361	0.526	0.606	0.524	0.622	0.641	1.041	1.238	0.683	0.624	0.556	0.702	0.651	0.924
rs over	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000
rs under	0.392	0.415	0.493	0.436	0.500	0.401	0.461	0.424	0.424	0.376	0.466	0.417	0.496	0.291	0.494	0.378	0.459	0.496	0.360	0.312	0.423
rs total	0.392	0.415	0.493	0.436	0.500	0.401	0.461	0.424	0.424	0.376	0.466	0.417	0.496	0.292	0.494	0.378	0.459	0.496	0.360	0.312	0.423

Figure 22: SOH results novel metrics LSTM

	VAH01	VAH02	VAH05	VAH06	VAH07	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	vah_total
crps	3.206	0.556	0.933	2.243	5.223	1.168	1.029	1.907	1.373	1.299	0.588	23.717	1.197	0.390	3.211	1.352	0.913	0.751	0.793	2.988	2.780
crps_W	1.983	0.506	1.233	1.451	2.612	0.727	1.255	2.635	1.942	1.870	0.735	23.124	1.748	0.336	2.159	1.118	0.600	0.444	0.592	3.963	2.760
cov - 0.5	0.562	0.750	0.276	0.474	0.000	0.545	0.333	0.156	0.368	0.300	0.500	1.000	0.273	0.800	0.615	0.733	0.500	0.545	0.500	0.294	0.452
mean width - 0.5	7.190	2.196	1.509	5.554	3.070	2.879	2.259	2.178	2.412	3.657	1.430	134.782	2.119	1.548	7.916	4.717	3.266	2.315	1.488	4.079	10.427
cov - 0.95	0.875	1.000	0.759	1.000	0.250	0.955	0.963	0.644	0.789	1.000	0.900	1.000	0.818	1.000	0.846	1.000	1.000	1.000	0.864	0.765	0.865
mean width - 0.95	20.811	6.147	4.359	16.413	9.091	8.524	6.471	6.303	7.070	10.720	4.176	390.314	6.122	4.498	22.934	13.624	9.485	6.932	4.320	11.914	30.235
rs over	0.070	0.183	0.000	0.010	0.000	0.026	0.002	0.000	0.001	0.053	0.064	0.457	0.000	0.167	0.099	0.120	0.118	0.079	0.001	0.000	0.005
rs under	0.021	0.003	0.190	0.017	0.479	0.011	0.114	0.272	0.136	0.094	0.043	0.000	0.151	0.001	0.010	0.001	0.000	0.004	0.057	0.188	0.043
rs total	0.091	0.186	0.190	0.027	0.479	0.037	0.116	0.272	0.137	0.148	0.108	0.457	0.151	0.169	0.109	0.120	0.118	0.083	0.058	0.188	0.048

Figure 23: SOH results novel metrics CNN mean stddev

	VAH01	VAH02	VAH05	VAH06	VAH07	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	vah_total
crps	2.007	2.824	0.423	1.768	8.847	1.925	0.437	0.354	1.338	0.585	0.828	23.499	0.430	0.386	4.688	0.321	0.416	0.446	0.744	0.507	2.381
crps_W	1.003	2.104	0.472	1.268	4.424	1.718	0.242	0.345	1.230	0.577	0.931	23.127	0.482	0.532	7.020	0.280	0.272	0.259	0.391	0.689	2.248
cov - 0.5	0.000	0.667	0.448	0.263	0.000	0.682	0.370	0.467	0.316	0.900	0.500	1.000	0.636	0.400	0.154	0.600	0.400	0.455	0.273	0.294	0.449
mean width - 0.5	0.925	8.329	0.928	0.926	0.909	5.638	0.925	0.919	0.930	2.504	2.752	136.462	0.942	0.960	0.921	0.931	0.915	0.941	0.927	0.933	9.144
cov - 0.95	0.000	1.000	0.966	0.474	0.000	0.955	0.926	0.956	0.579	1.000	1.000	1.000	0.818	1.000	0.154	1.000	1.000	0.818	0.682	0.882	0.801
mean width - 0.95	2.694	24.416	2.705	2.703	2.691	16.582	2.729	2.703	2.675	7.057	8.021	392.564	2.713	2.715	2.725	2.714	2.737	2.699	2.690	2.701	26.371
rs over	0.000	0.064	0.004	0.000	0.000	0.143	0.003	0.034	0.000	0.196	0.084	0.479	0.046	0.020	0.000	0.060	0.005	0.004	0.000	0.001	0.006
rs under	0.495	0.002	0.045	0.273	0.500	0.001	0.076	0.024	0.216	0.000	0.002	0.000	0.028	0.058	0.382	0.002	0.069	0.054	0.162	0.131	0.057
rs total	0.495	0.066	0.049	0.273	0.500	0.144	0.079	0.058	0.216	0.196	0.087	0.479	0.074	0.077	0.382	0.062	0.074	0.058	0.162	0.131	0.063

Figure 24: SOH results novel metrics LSTM mean stddev

	VAH01	VAH02	VAH05	VAH06	VAH07	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	vah_total
crps	2.029	0.268	1.031	0.582	8.178	1.221	0.657	1.934	1.266	0.276	0.233	0.647	0.201	0.378	2.699	0.291	0.600	0.473	0.821	0.820	1.088
crps_W	1.015	0.342	1.380	0.653	4.089	1.741	0.612	2.626	1.563	0.291	0.158	0.963	0.208	0.450	3.589	0.278	0.343	0.328	0.605	1.038	1.203
cov - 0.5	0.000	0.333	0.138	0.474	0.000	0.136	0.296	0.044	0.000	0.100	0.600	0.053	0.636	0.600	0.077	0.133	0.400	0.273	0.045	0.059	0.185
mean width - 0.5	0.684	0.488	0.332	0.506	0.500	0.439	0.617	0.679	0.806	0.601	0.557	0.627	0.648	0.747	0.645	0.391	0.669	0.674	0.524	0.459	0.575
cov - 0.95	0.188	0.750	0.379	0.632	0.000	0.364	0.556	0.178	0.368	0.900	1.000	0.421	1.000	0.900	0.231	0.667	0.400	0.636	0.409	0.294	0.463
mean width - 0.95	1.954	1.474	0.953	1.451	1.454	1.272	1.813	1.973	2.362	1.728	1.641	1.793	1.918	2.194	1.862	1.139	2.210	2.022	1.513	1.346	1.681
rs over	0.000	0.004	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.069	0.001	0.133	0.127	0.003	0.000	0.026	0.001	0.001	0.000	0.000
rs under	0.463	0.079	0.347	0.129	0.500	0.324	0.226	0.425	0.418	0.219	0.027	0.350	0.000	0.009	0.395	0.232	0.168	0.182	0.365	0.379	0.280
rs total	0.463	0.083	0.347	0.130	0.500	0.325	0.226	0.425	0.418	0.220	0.096	0.351	0.133	0.136	0.397	0.232	0.195	0.183	0.366	0.379	0.280

Figure 25: SOH results novel metrics MDN

	VAH01	VAH02	VAH05	VAH06	VAH07	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	vah_total
crps	2.392	4.234	3.269	3.218	4.555	2.989	3.599	2.810	3.426	3.328	2.813	3.091	3.030	3.176	2.897	2.808	3.257	3.113	3.028	3.017	3.130
crps_W	2.549	2.545	3.440	3.294	2.316	2.394	4.353	3.641	3.208	2.710	1.818	2.662	2.188	2.098	1.823	2.191	2.590	2.000	2.654	3.096	2.916
cov - 0.5	0.562	0.667	0.586	0.632	0.000	0.500	0.407	0.556	0.474	0.800	0.500	0.579	0.364	0.300	0.538	0.667	0.400	0.545	0.682	0.588	0.543
mean width - 0.5	7.087	14.022	10.641	11.552	8.160	9.058	7.861	9.671	8.412	11.954	5.947	9.617	6.138	4.729	6.765	8.010	7.529	6.344	9.388	9.669	8.943
cov - 0.95	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.800	1.000	0.909	0.800	0.846	0.933	1.000	0.818	1.000	1.000	0.971
mean width - 0.95	20.794	39.944	30.558	34.163	23.801	26.192	22.805	28.041	24.469	34.846	17.246	27.954	17.576	13.571	19.391	23.461	21.667	18.612	27.390	28.084	25.946
rs over	0.066	0.077	0.082	0.123	0.004	0.054	0.007	0.095	0.021	0.147	0.009	0.087	0.001	0.000	0.065	0.065	0.019	0.023	0.080	0.067	0.035
rs under	0.001	0.009	0.000	0.000	0.244	0.007	0.063	0.040	0.016	0.000	0.040	0.001	0.097	0.136	0.031	0.005	0.082	0.043	0.005	0.002	0.001
rs total	0.067	0.086	0.082	0.124	0.248	0.062	0.069	0.136	0.037	0.147	0.050	0.088	0.098	0.136	0.096	0.070	0.100	0.066	0.085	0.069	0.036

Figure 26: SOH results novel metrics BNN

10.5 Appendix A-5: RUL results generic metrics

	VAH01	VAH02	VAH05	VAH06	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	VAH_ALL
mae	111.51	61.56	42.12	114.66	141.15	65.53	76.33	24.43	14.27	52.81	32.34	32.14	14.17	89.81	57.19	121.22	39.97	76.28	164.35	70.10
rmse	129.46	82.10	56.27	119.05	174.19	70.40	90.67	29.13	17.39	62.89	36.33	39.61	17.94	105.91	61.58	153.88	47.38	84.06	165.54	81.25

Figure 27: RUL results MAE RMSE CNN

	VAH01	VAH02	VAH05	VAH06	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	VAH_ALL
mae	107.42	83.97	57.97	84.24	140.14	61.22	74.17	71.08	12.42	24.17	45.78	20.44	7.21	330.18	108.89	135.8	40.89	25.56	111.90	81.23
rmse	117.58	92.47	65.40	90.61	168.34	66.57	82.58	74.20	18.23	29.02	51.26	29.51	11.02	352.79	111.21	178.6	45.44	34.46	112.96	91.17

Figure 28: RUL results MAE RMSE LSTM

	VAH01	VAH02	VAH05	VAH06	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	VAH_ALL
mae	69.74	39.91	70.36	144.12	190.04	71.60	106.38	79.28	32.58	24.94	52.69	43.80	30.35	138.48	56.17	115.57	79.92	22.54	139.43	79.36
rmse	83.07	46.75	85.91	155.57	227.95	80.17	124.79	94.60	38.71	35.37	61.30	50.83	33.24	164.01	73.12	139.02	91.17	27.96	141.78	92.39

Figure 29: RUL results MAE RMSE CNN mean stddev

	VAH01	VAH02	VAH05	VAH06	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	VAH_ALL
mae	100.82	56.10	66.18	95.13	137.36	93.42	126.23	19.8	33.6	22.98	22.16	19.63	22.05	115.86	75.56	141.19	28.87	30.86	105.27	69.11
rmse	110.39	66.95	78.11	98.80	167.27	99.30	145.62	26.5	39.6	28.01	25.82	21.68	25.88	139.89	83.76	181.01	34.29	36.77	105.46	79.74

Figure 30: RUL results MAE RMSE LSTM mean stddev

	VAH01	VAH02	VAH05	VAH06	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	VAH_ALL
mae	119.90	67.36	40.17	93.95	127.15	67.07	81.66	31.59	12.93	30.35	32.00	20.31	8.83	118.36	110.13	110.22	44.43	33.00	117.14	66.66
rmse	140.83	78.52	44.65	96.39	158.14	72.75	89.68	34.04	16.93	32.92	35.78	25.54	12.31	141.39	115.21	139.50	49.56	37.59	126.95	76.25

Figure 31: RUL results MAE RMSE MDN

	VAH01	VAH02	VAH05	VAH06	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	VAH_ALL
mae	150.00	125.32	205.33	341.06	197.81	152.09	199.19	162.78	132.22	134.66	142.04	125.34	115.45	140.04	149.13	129.91	145.01	185.76	139.56	161.72
rmse	172.36	146.46	240.99	388.67	235.67	176.15	234.48	188.38	157.50	161.18	160.75	151.92	138.19	160.54	171.54	152.39	167.97	214.17	164.42	188.62

Figure 32: RUL results MAE RMSE BNN

10.6 Appendix A-6: RUL results Novel metrics

	VAH01	VAH02	VAH05	VAH06	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	vah_total
crps	100.534	54.552	32.300	104.469	136.354	54.799	67.954	17.348	9.606	44.649	24.707	24.404	10.081	79.037	47.551	116.548	29.310	66.350	153.293	64.093
crps_W	50.268	79.196	43.908	52.237	82.990	34.991	35.738	21.764	7.372	66.554	18.964	24.612	11.230	98.640	71.276	146.500	37.450	35.557	229.940	59.434
cov - 0.5	0.000	0.200	0.267	0.000	0.000	0.083	0.067	0.333	0.333	0.222	0.182	0.333	0.333	0.091	0.091	0.000	0.200	0.071	0.000	0.138
mean width - 0.5	27.435	21.024	31.467	24.313	11.694	26.843	22.997	29.455	23.005	23.949	23.802	24.912	13.719	29.519	24.015	11.215	32.823	31.951	26.268	24.400
cov - 0.95	0.083	0.400	0.733	0.077	0.067	0.083	0.267	1.000	1.000	0.333	0.727	0.667	0.778	0.273	0.182	0.100	0.700	0.286	0.000	0.392
mean width - 0.95	77.571	62.372	92.659	72.248	34.474	79.074	67.343	87.823	69.727	69.363	68.671	76.213	41.386	89.083	71.568	36.168	96.669	92.867	77.823	72.135
rs over	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.002	0.011	0.002	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
rs under	0.493	0.319	0.197	0.487	0.483	0.435	0.378	0.133	0.086	0.278	0.313	0.218	0.184	0.363	0.428	0.458	0.240	0.416	0.500	0.345
rs total	0.493	0.319	0.198	0.487	0.483	0.435	0.378	0.135	0.097	0.280	0.313	0.218	0.184	0.363	0.428	0.458	0.240	0.416	0.500	0.345

Figure 33: RUL results novel metrics CNN

	VAH01	VAH02	VAH05	VAH06	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	vah_total
crps	107.992	81.557	53.965	82.313	138.988	58.128	71.625	65.917	8.803	22.272	43.064	17.677	6.691	322.334	105.657	132.753	39.486	23.120	110.368	80.503
crps_W	53.996	114.766	69.352	41.157	85.714	29.067	37.626	98.875	9.791	33.407	49.504	17.657	5.776	483.502	158.486	177.124	38.819	22.743	165.552	87.999
cov - 0.5	0.000	0.000	0.000	0.000	0.000	0.083	0.067	0.000	0.111	0.000	0.000	0.222	0.333	0.000	0.000	0.000	0.000	0.071	0.000	0.041
mean width - 0.5	7.902	3.502	8.402	4.766	2.763	8.166	6.349	12.124	7.439	4.527	6.065	9.390	2.613	18.070	6.798	7.723	3.251	7.069	3.328	6.894
cov - 0.95	0.000	0.100	0.133	0.000	0.000	0.083	0.067	0.000	0.333	0.111	0.091	0.333	0.556	0.000	0.000	0.000	0.100	0.214	0.000	0.101
mean width - 0.95	19.744	21.528	31.409	13.387	7.922	22.227	17.930	34.549	29.470	13.076	19.945	26.295	8.180	55.924	24.235	23.602	10.154	22.271	11.266	21.764
rs over	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000
rs under	0.500	0.460	0.465	0.500	0.500	0.421	0.437	0.500	0.366	0.464	0.487	0.338	0.149	0.500	0.500	0.498	0.487	0.433	0.500	0.453
rs total	0.500	0.460	0.465	0.500	0.500	0.422	0.437	0.500	0.366	0.464	0.487	0.338	0.150	0.500	0.500	0.498	0.487	0.433	0.500	0.453

Figure 34: RUL results novel metrics LSTM

	VAH01	VAH02	VAH05	VAH06	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	vah_total
crps	52.066	26.703	48.506	105.910	152.382	47.598	81.821	54.468	24.069	27.033	39.399	33.016	30.383	114.298	40.007	81.841	54.558	19.185	101.324	62.402
crps_W	42.821	21.439	56.294	53.322	89.524	29.259	43.497	62.276	32.897	32.373	30.992	43.832	30.716	68.321	57.122	111.199	42.778	19.145	151.966	53.761
cov - 0.5	0.667	0.500	0.333	0.077	0.133	0.417	0.200	0.583	0.778	0.889	0.636	0.889	0.889	0.091	0.545	0.400	0.500	0.786	0.000	0.465
mean width - 0.5	174.439	76.796	87.964	109.804	115.158	123.993	92.340	156.801	90.610	124.966	124.301	139.619	145.345	69.671	97.247	213.237	171.848	82.376	84.381	118.186
cov - 0.95	0.917	1.000	0.933	0.615	0.467	1.000	0.600	1.000	1.000	1.000	1.000	1.000	1.000	0.364	0.909	1.000	1.000	1.000	0.400	0.839
mean width - 0.95	500.211	227.757	255.431	319.021	336.812	364.878	269.178	466.974	267.187	361.706	363.545	414.946	430.402	200.205	280.688	623.627	498.024	246.148	243.981	345.577
rs over	0.073	0.049	0.002	0.000	0.000	0.005	0.002	0.040	0.142	0.274	0.069	0.191	0.241	0.000	0.028	0.044	0.052	0.226	0.000	0.000
rs under	0.004	0.031	0.109	0.354	0.309	0.069	0.257	0.014	0.000	0.000	0.013	0.003	0.002	0.356	0.016	0.037	0.018	0.000	0.452	0.049
rs total	0.077	0.080	0.111	0.354	0.309	0.074	0.258	0.054	0.142	0.274	0.083	0.194	0.243	0.356	0.044	0.080	0.069	0.226	0.452	0.050

Figure 35: RUL results novel metrics CNN mean stddev

	VAH01	VAH02	VAH05	VAH06	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	vah_total
crps	92.637	55.723	57.746	70.354	128.223	88.600	93.476	28.805	21.686	32.770	22.029	13.918	15.990	99.907	52.316	119.249	24.730	24.708	83.177	62.108
crps_W	46.338	78.081	59.011	35.192	78.364	44.300	48.853	32.643	31.199	29.009	22.180	12.367	10.450	54.831	63.587	164.142	26.157	18.357	124.749	51.670
cov - 0.5	0.083	0.000	0.867	0.000	0.067	0.000	0.200	0.833	0.444	1.000	0.727	0.778	0.444	0.000	0.364	0.000	0.700	0.143	0.000	0.336
mean width - 0.5	20.789	0.914	255.328	63.129	23.779	11.576	117.422	138.407	61.110	169.622	90.112	50.294	31.030	42.282	124.248	55.648	97.802	26.248	49.109	77.657
cov - 0.95	0.083	0.100	1.000	0.538	0.200	0.000	0.667	1.000	1.000	1.000	0.909	1.000	0.889	0.364	0.818	0.400	1.000	0.571	0.200	0.604
mean width - 0.95	59.372	2.688	735.972	184.807	68.773	33.703	344.943	403.048	174.451	508.887	261.483	145.677	89.376	122.153	366.421	162.051	287.626	77.598	144.263	226.590
rs over	0.000	0.000	0.203	0.000	0.000	0.000	0.000	0.270	0.041	0.333	0.137	0.103	0.001	0.000	0.002	0.000	0.139	0.000	0.000	0.000
rs under	0.434	0.489	0.001	0.440	0.403	0.500	0.219	0.000	0.037	0.000	0.002	0.005	0.087	0.396	0.143	0.388	0.001	0.259	0.470	0.171
rs total	0.434	0.489	0.203	0.440	0.403	0.500	0.219	0.270	0.079	0.333	0.140	0.108	0.088	0.396	0.145	0.388	0.139	0.259	0.470	0.171

Figure 36: RUL results novel metrics LSTM mean stddev

	VAH01	VAH02	VAH05	VAH06	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	vah_total
crps	117.220	58.194	34.463	90.073	122.818	61.885	73.943	21.821	11.524	23.554	26.449	16.827	7.089	108.488	104.611	104.392	38.100	22.597	110.561	62.881
crps_W	58.756	82.297	46.402	45.037	74.704	30.943	45.609	30.556	6.832	35.153	28.574	9.047	9.255	75.985	156.917	138.047	33.840	20.768	165.842	57.323
cov - 0.5	0.083	0.000	0.067	0.000	0.067	0.000	0.000	0.167	0.222	0.111	0.182	0.222	0.333	0.091	0.000	0.000	0.000	0.357	0.000	0.097
mean width - 0.5	8.488	23.964	17.609	9.237	10.958	15.492	19.236	34.914	10.936	22.662	26.548	11.525	11.571	25.621	13.202	14.101	15.076	48.474	15.973	19.092
cov - 0.95	0.083	0.200	0.200	0.000	0.067	0.083	0.133	0.750	0.667	0.556	0.364	0.556	0.667	0.182	0.000	0.000	0.200	0.929	0.000	0.286
mean width - 0.95	23.282	61.820	50.208	26.914	31.757	49.678	55.392	101.525	33.266	62.994	81.887	37.522	33.590	75.273	38.365	40.586	44.660	142.333	45.345	55.659
rs over	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.004	0.000	0.014	0.000	0.000	0.000	0.000	0.001	0.000	0.000
rs under	0.443	0.462	0.412	0.500	0.443	0.492	0.480	0.232	0.268	0.303	0.281	0.317	0.121	0.405	0.500	0.497	0.449	0.102	0.499	0.382
rs total	0.443	0.462	0.412	0.500	0.443	0.492	0.480	0.232	0.268	0.303	0.284	0.317	0.135	0.405	0.500	0.497	0.449	0.102	0.499	0.382

Figure 37: RUL results novel metrics MDN

	VAH01	VAH02	VAH05	VAH06	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	vah_total
crps	99.396	85.127	144.054	230.809	139.572	103.857	139.835	109.482	90.774	91.749	93.722	86.844	80.253	93.739	100.599	88.493	96.291	126.018	94.171	114.403
crps_W	98.781	100.697	114.733	121.549	103.571	103.611	109.710	86.943	117.519	118.084	102.147	111.818	100.425	103.340	110.336	106.923	113.518	105.741	113.870	107.408
cov - 0.5	0.417	0.600	0.267	0.308	0.333	0.500	0.333	0.417	0.556	0.556	0.455	0.556	0.556	0.545	0.364	0.500	0.400	0.357	0.400	0.429
mean width - 0.5	249.303	249.546	249.346	443.092	244.960	253.285	250.511	234.643	252.442	258.637	253.852	244.728	232.490	255.424	253.364	256.301	257.304	252.571	254.326	261.630
cov - 0.95	1.000	1.000	0.867	0.923	0.867	1.000	0.867	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.929	1.000	0.963
mean width - 0.95	738.136	720.095	716.481	1276.771	703.181	739.604	730.506	690.129	742.591	747.794	740.367	705.810	675.100	738.384	742.220	743.938	750.861	737.524	743.482	760.582
rs over	0.012	0.044	0.000	0.001	0.001	0.008	0.001	0.002	0.034	0.048	0.022	0.046	0.043	0.023	0.014	0.044	0.022	0.000	0.031	0.002
rs under	0.042	0.011	0.140	0.133	0.126	0.040	0.119	0.092	0.010	0.007	0.032	0.005	0.006	0.024	0.045	0.011	0.047	0.104	0.016	0.045
rs total	0.055	0.054	0.140	0.134	0.127	0.048	0.120	0.093	0.043	0.055	0.054	0.051	0.048	0.046	0.059	0.054	0.069	0.104	0.047	0.047

Figure 38: RUL results novel metrics BNN

10.7 Appendix A-7: RUC results generic metrics

	VAH01	VAH02	VAH05	VAH06	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	VAH_ALL
mae	2.09	1.58	0.98	1.50	2.70	1.64	1.39	0.6	0.08	0.84	1.11	0.38	0.20	1.95	1.58	2.25	0.63	1.02	2.67	1.33
rmse	2.25	1.98	1.15	1.66	3.31	1.77	1.68	0.7	0.11	1.00	1.25	0.56	0.25	2.35	1.69	2.82	0.66	1.24	2.73	1.53

Figure 39: RUC results MAE RMSE CNN

	VAH01	VAH02	VAH05	VAH06	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	VAH_ALL
mae	1.76	1.51	1.16	1.80	2.81	1.16	1.81	1.34	0.19	0.41	0.78	0.21	0.23	1.85	1.89	2.43	0.84	0.60	2.08	1.31
rmse	2.14	1.66	1.44	1.87	3.40	1.36	2.34	1.44	0.25	0.52	0.85	0.32	0.32	2.22	1.94	3.15	0.92	0.71	2.09	1.52

Figure 40: RUC results MAE RMSE LSTM

	VAH01	VAH02	VAH05	VAH06	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	VAH_ALL
mae	2.58	1.64	1.46	1.48	2.78	1.72	1.33	1.23	0.85	0.69	0.71	0.41	2.65	2.76	1.17	2.46	0.70	1.24	2.54	1.60
rmse	2.83	1.92	1.67	1.71	3.48	2.32	1.67	1.43	1.05	0.80	0.83	0.47	3.26	3.19	1.60	3.35	1.09	1.65	2.54	1.94

Figure 41: RUC results MAE RMSE CNN mean stddev

	VAH01	VAH02	VAH05	VAH06	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	VAH_ALL
mae	2.30	1.85	0.83	1.61	2.62	1.68	1.17	1.58	0.16	0.36	1.20	0.57	0.16	2.98	1.63	2.59	0.92	0.75	2.01	1.42
rmse	2.61	2.13	0.94	1.69	3.16	1.75	1.33	1.88	0.21	0.38	1.26	0.64	0.23	3.22	1.71	3.31	0.98	0.88	2.02	1.60

Figure 42: RUC results MAE RMSE LSTM mean stddev

	VAH01	VAH02	VAH05	VAH06	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	VAH_ALL
mae	2.21	0.95	0.60	1.91	2.34	1.43	2.67	1.82	0.25	0.93	0.64	0.65	0.18	2.05	1.8	2.25	0.76	0.22	3.68	1.44
rmse	2.66	1.08	0.73	2.00	2.87	1.52	3.00	1.96	0.38	0.94	0.72	0.71	0.28	2.50	1.9	3.05	0.84	0.27	3.70	1.64

Figure 43: RUC results MAE RMSE MDN

	VAH01	VAH02	VAH05	VAH06	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	VAH_ALL
mae	2.96	2.61	7.02	6.59	7.35	5.54	3.93	3.09	2.52	4.03	5.04	2.57	2.46	5.03	4.95	2.71	2.60	3.61	2.60	4.06
rmse	3.43	3.05	8.25	7.58	8.58	6.53	4.61	3.61	3.02	4.79	5.89	3.11	2.95	5.94	5.87	3.15	3.05	4.20	3.08	4.77

Figure 44: RUC results MAE RMSE BNN

10.8 Appendix A-8: RUC results Novel metrics

	VAH01	VAH02	VAH05	VAH06	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	vah_total
crps	1.909	1.400	0.874	1.399	2.601	1.468	1.259	0.446	0.051	0.722	0.972	0.301	0.140	1.657	1.480	2.098	0.507	0.939	2.531	1.243
crps_W	0.955	2.065	1.186	0.700	1.559	0.791	0.651	0.609	0.072	1.081	0.803	0.301	0.132	2.041	2.220	2.605	0.514	0.489	3.797	1.175
cov - 0.5	0.000	0.000	0.133	0.000	0.067	0.167	0.133	0.167	0.444	0.222	0.000	0.444	0.333	0.182	0.000	0.000	0.000	0.071	0.000	0.115
mean width - 0.5	0.409	0.390	0.268	0.258	0.246	0.727	0.411	0.521	0.138	0.315	0.322	0.490	0.249	0.742	0.234	0.372	0.295	0.257	0.326	0.367
cov - 0.95	0.000	0.200	0.267	0.000	0.067	0.250	0.333	0.667	1.000	0.222	0.091	0.889	0.889	0.364	0.000	0.100	0.100	0.214	0.000	0.276
mean width - 0.95	1.252	1.192	0.841	0.741	0.717	1.816	1.119	1.459	0.453	0.914	0.916	1.538	0.842	2.142	0.738	1.097	0.824	0.753	1.003	1.066
rs over	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.034	0.000	0.000	0.003	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000
rs under	0.497	0.478	0.385	0.500	0.451	0.358	0.333	0.246	0.051	0.333	0.487	0.061	0.116	0.358	0.500	0.489	0.474	0.423	0.500	0.378
rs total	0.497	0.478	0.385	0.500	0.451	0.358	0.333	0.246	0.086	0.333	0.487	0.064	0.116	0.358	0.500	0.489	0.474	0.423	0.500	0.378

Figure 45: RUC results novel metrics CNN

	VAH01	VAH02	VAH05	VAH06	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	vah_total
crps	1.715	1.459	1.108	1.769	2.766	1.127	1.774	1.276	0.159	0.386	0.746	0.171	0.200	1.635	1.812	2.386	0.810	0.564	2.033	1.319
crps_W	0.857	2.003	1.581	0.885	1.724	0.564	0.993	1.913	0.159	0.555	0.794	0.141	0.297	1.919	2.718	3.049	0.808	0.469	3.049	1.298
cov - 0.5	0.000	0.000	0.000	0.000	0.000	0.083	0.000	0.000	0.111	0.111	0.000	0.000	0.222	0.000	0.000	0.000	0.000	0.143	0.000	0.032
mean width - 0.5	0.097	0.104	0.131	0.076	0.094	0.084	0.089	0.157	0.082	0.090	0.078	0.110	0.088	0.574	0.193	0.081	0.082	0.111	0.108	0.128
cov - 0.95	0.000	0.000	0.067	0.000	0.000	0.167	0.133	0.000	0.333	0.111	0.000	0.444	0.333	0.182	0.000	0.000	0.000	0.143	0.000	0.092
mean width - 0.95	0.283	0.332	0.378	0.228	0.278	0.260	0.257	0.471	0.238	0.259	0.235	0.301	0.257	1.589	0.546	0.303	0.240	0.308	0.362	0.374
rs over	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.006	0.000	0.000	0.001	0.000	0.000	0.000	0.000	0.000	0.000	0.000
rs under	0.500	0.500	0.473	0.500	0.500	0.423	0.468	0.500	0.399	0.391	0.500	0.380	0.276	0.424	0.500	0.500	0.499	0.394	0.500	0.458
rs total	0.500	0.500	0.473	0.500	0.500	0.423	0.468	0.500	0.399	0.397	0.500	0.380	0.277	0.424	0.500	0.500	0.499	0.394	0.500	0.458

Figure 46: RUC results novel metrics LSTM

	VAH01	VAH02	VAH05	VAH06	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	vah_total
crps	2.236	1.260	1.035	1.065	2.473	1.411	1.104	0.912	0.607	0.474	0.680	0.282	1.857	1.865	0.900	2.172	0.591	0.878	2.148	1.284
crps_W	1.125	1.886	1.280	0.621	1.274	1.555	0.690	1.284	0.846	0.560	0.633	0.271	2.476	2.052	1.252	2.868	0.749	0.925	3.222	1.314
cov - 0.5	0.083	0.100	0.200	0.231	0.200	0.333	0.333	0.083	0.556	0.333	0.818	0.556	0.556	0.455	0.455	0.200	0.800	0.500	0.000	0.346
mean width - 0.5	0.925	1.071	1.412	1.415	0.939	1.300	0.925	0.922	1.639	1.074	2.960	0.940	4.765	5.163	1.303	0.923	1.536	2.361	0.928	1.670
cov - 0.95	0.167	0.500	0.733	0.846	0.400	0.667	0.600	0.750	1.000	1.000	1.000	1.000	1.000	1.000	0.818	0.400	0.900	1.000	0.000	0.714
mean width - 0.95	2.708	3.103	3.965	4.123	2.705	3.781	2.668	2.724	4.809	3.111	8.799	2.670	13.797	14.774	3.872	2.680	4.383	6.857	2.688	4.839
rs over	0.001	0.000	0.000	0.000	0.000	0.002	0.000	0.000	0.031	0.012	0.183	0.085	0.029	0.028	0.019	0.000	0.148	0.043	0.000	0.000
rs under	0.407	0.335	0.244	0.240	0.308	0.197	0.215	0.303	0.011	0.082	0.005	0.018	0.012	0.017	0.085	0.295	0.014	0.002	0.500	0.153
rs total	0.408	0.335	0.244	0.240	0.308	0.199	0.215	0.303	0.042	0.094	0.188	0.103	0.040	0.046	0.104	0.295	0.162	0.045	0.500	0.153

Figure 47: RUC results novel metrics CNN mean stddev

	VAH01	VAH02	VAH05	VAH06	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	vah_total
crps	1.999	1.541	0.567	1.255	2.282	1.378	0.910	1.264	0.188	0.243	0.856	0.371	0.190	2.603	1.228	2.284	0.620	0.525	1.531	1.180
crps_W	1.005	2.209	0.806	0.628	1.383	0.762	0.486	1.887	0.164	0.339	0.815	0.285	0.198	3.905	1.842	3.066	0.602	0.288	2.296	1.193
cov - 0.5	0.083	0.100	0.200	0.000	0.133	0.083	0.133	0.167	0.889	0.889	0.000	0.444	0.889	0.000	0.000	0.100	0.100	0.286	0.000	0.212
mean width - 0.5	0.940	0.917	0.931	0.940	0.939	0.938	0.934	0.917	0.933	0.938	0.930	0.920	0.904	0.928	0.933	0.937	0.934	0.929	0.941	0.931
cov - 0.95	0.167	0.400	0.800	0.154	0.267	0.167	0.667	0.500	1.000	1.000	0.727	1.000	1.000	0.091	0.364	0.400	1.000	0.929	0.000	0.544
mean width - 0.95	2.697	2.746	2.755	2.709	2.683	2.693	2.722	2.717	2.623	2.663	2.723	2.688	2.722	2.707	2.694	2.667	2.681	2.684	2.673	2.699
rs over	0.000	0.004	0.000	0.000	0.000	0.000	0.000	0.003	0.323	0.142	0.000	0.009	0.304	0.000	0.000	0.000	0.004	0.001	0.000	0.000
rs under	0.406	0.327	0.180	0.439	0.364	0.425	0.285	0.306	0.000	0.032	0.373	0.061	0.000	0.469	0.445	0.341	0.272	0.132	0.494	0.257
rs total	0.406	0.330	0.180	0.439	0.364	0.425	0.285	0.309	0.323	0.174	0.373	0.070	0.304	0.469	0.445	0.341	0.277	0.132	0.494	0.257

Figure 48: RUC results novel metrics LSTM mean stddev

	VAH01	VAH02	VAH05	VAH06	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	vah_total
crps	2.120	0.789	0.497	1.858	2.179	1.369	2.523	1.478	0.205	0.772	0.538	0.535	0.120	1.973	1.689	2.088	0.830	0.197	3.636	1.377
crps_W	1.061	1.077	0.593	0.929	1.341	0.685	1.446	1.975	0.113	1.157	0.638	0.272	0.157	1.298	2.534	2.857	0.737	0.185	5.455	1.271
cov - 0.5	0.000	0.000	0.200	0.000	0.000	0.000	0.000	0.000	0.556	0.000	0.182	0.111	0.667	0.091	0.000	0.000	0.100	0.357	0.000	0.111
mean width - 0.5	0.180	0.268	0.334	0.163	0.232	0.157	0.204	1.017	0.199	0.201	0.564	0.360	0.225	0.194	0.260	0.516	0.310	0.257	0.106	0.302
cov - 0.95	0.167	0.200	0.333	0.154	0.067	0.083	0.067	0.333	0.778	0.111	0.364	0.333	0.778	0.182	0.000	0.300	0.300	0.714	0.000	0.267
mean width - 0.95	0.768	0.733	0.934	0.860	0.660	0.569	0.630	2.979	0.564	0.808	1.661	1.100	0.651	0.588	0.767	1.331	1.550	0.918	0.304	0.966
rs over	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.006	0.000	0.000	0.000	0.087	0.000	0.000	0.000	0.000	0.000	0.000	0.000
rs under	0.451	0.456	0.282	0.479	0.469	0.487	0.490	0.415	0.113	0.486	0.331	0.352	0.044	0.430	0.500	0.390	0.353	0.141	0.500	0.378
rs total	0.451	0.456	0.282	0.479	0.469	0.487	0.490	0.415	0.119	0.486	0.331	0.352	0.131	0.430	0.500	0.390	0.353	0.141	0.500	0.378

Figure 49: RUC results novel metrics MDN

	VAH01	VAH02	VAH05	VAH06	VAH09	VAH10	VAH12	VAH13	VAH15	VAH16	VAH17	VAH20	VAH22	VAH23	VAH24	VAH25	VAH27	VAH28	VAH30	vah_total
crps	1.493	1.155	1.443	1.702	2.604	1.583	1.757	1.070	1.303	1.626	1.236	0.985	0.939	2.685	0.992	1.650	1.989	2.392	0.614	1.586
crps_W	1.269	1.677	1.578	0.920	2.597	1.038	1.276	1.247	1.608	2.312	1.386	1.386	0.993	2.132	1.213	2.194	2.810	2.911	0.608	1.653
cov - 0.5	0.667	0.500	0.533	0.308	0.667	0.333	0.267	0.500	0.222	0.222	0.818	0.556	0.667	0.455	0.727	0.800	0.600	0.857	0.800	0.553
mean width - 0.5	5.364	3.281	4.653	2.923	8.574	3.269	3.195	2.936	3.123	2.926	5.306	2.691	2.837	5.382	3.937	6.144	6.853	10.684	2.511	4.727
cov - 0.95	1.000	1.000	0.867	0.923	0.733	1.000	0.933	1.000	0.889	1.000	1.000	1.000	0.889	0.818	1.000	1.000	1.000	1.000	1.000	0.945
mean width - 0.95	15.447	9.435	13.579	8.541	25.273	9.378	9.282	8.676	8.988	8.496	14.982	7.748	8.042	15.500	11.672	18.389	20.004	31.209	7.181	13.757
rs over	0.105	0.035	0.043	0.001	0.102	0.004	0.000	0.049	0.000	0.005	0.220	0.033	0.035	0.010	0.155	0.147	0.106	0.244	0.227	0.034
rs under	0.003	0.065	0.019	0.156	0.036	0.076	0.116	0.009	0.119	0.144	0.006	0.028	0.053	0.085	0.006	0.007	0.014	0.000	0.000	0.002
rs total	0.108	0.100	0.062	0.157	0.137	0.079	0.116	0.058	0.119	0.149	0.227	0.061	0.088	0.095	0.161	0.154	0.120	0.244	0.227	0.036

Figure 50: RUC results novel metrics BNN

References

[LDS18] Xiang Li, Qian Ding, and Jian-Qiao Sun. "Remaining useful life estimation in prognostics using deep convolution neural networks". In: *Reliability Engineering System Safety* 172 (2018), pp. 1–11. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2017.11.021. URL: https://www.sciencedirect.com/science/article/pii/S0951832017307779.

[NM19] Khanh T.P. Nguyen and Kamal Medjaher. "A new dynamic predictive maintenance framework using deep learning for failure prognostics". In: *Reliability Engineering System Safety* 188 (2019), pp. 251–262. ISSN: 0951-8320. DOI: https://doi.org/10.1016/j.ress.2019.03.018. URL: https://www.sciencedirect.com/science/article/pii/S0951832018311050.

[Pol+19] Nicholas Polaczyk et al. “A review of current technology and research in urban on-demand air mobility applications”. English. In: *8th Biennial Autonomous VTOL Technical Meeting and 6th Annual Electric VTOL Symposium 2019*. 8th Biennial Autonomous VTOL Technical Meeting and 6th Annual Electric VTOL Symposium 2019 ; Conference date: 28-01-2019 Through 01-02-2019. Vertical Flight Society, 2019, pp. 333–343.

[Orr+20] Pier Francesco Orrù et al. “Machine Learning Approach Using MLP and SVM Algorithms for the Fault Prediction of a Centrifugal Pump in the Oil and Gas Industry”. In: *Sustainability* 12.11 (2020). ISSN: 2071-1050. DOI: 10.3390/su12114776. URL: https://www.mdpi.com/2071-1050/12/11/4776.

[Ben+21] Maximilian Benker et al. “Utilizing uncertainty information in remaining useful life estimation via Bayesian neural networks and Hamiltonian Monte Carlo”. In: *Journal of Manufacturing Systems* 61 (2021), pp. 799–807. ISSN: 0278-6125. DOI: https://doi.org/10.1016/j.jmsy.2020.11.005. URL: https://www.sciencedirect.com/science/article/pii/S0278612520301928.

[Kim21] Taewoon Kim. “Generalizing MLPs With Dropouts, Batch Normalization, and Skip Connections”. In: *CoRR* abs/2108.08186 (2021). arXiv: 2108.08186. URL: https://arxiv.org/abs/2108.08186.

[Nil+21] Viktor Nilsson et al. “Probabilistic dose prediction using mixture density networks for automated radiation therapy treatment planning”. In: *Physics in Medicine Biology* 66.5 (Feb. 2021), p. 055003. DOI: 10.1088/1361-6560/abdd8a. URL: https://dx.doi.org/10.1088/1361-6560/abdd8a.

[Che+22] Yunhong Che et al. “Data efficient health prognostic for batteries based on sequential information-driven probabilistic neural network”. In: *Applied Energy* 323 (2022), p. 119663. ISSN: 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2022.119663. URL: https://www.sciencedirect.com/science/article/pii/S0306261922009618.

[PM22] Ingeborg de Pater and Mihaela Mitici. “Novel Metrics to Evaluate Probabilistic Remaining Useful Life Prognostics with Applications to Turbofan Engines”. In: *PHM Society European Conference* 7 (June 2022), pp. 96–109. DOI: 10.36001/phme.2022.v7i1.3320.

[Rem+22] I. Remadna et al. “Boosting RUL Prediction Using a Hybrid Deep CNN-BLSTM Architecture”. In: *Automatic Control and Computer Sciences* 56.4 (Aug. 2022), pp. 300–310. ISSN: 1558-108X. DOI: 10.3103/S014641162204006X. URL: https://doi.org/10.3103/S014641162204006X.

[RDH22] Andy Rivas, Gregory Kyriakos Delipei, and Jason Hou. “Predictions of component Remaining Useful Lifetime Using Bayesian Neural Network”. In: *Progress in Nuclear Energy* 146 (2022), p. 104143. ISSN: 0149-1970. DOI: https://doi.org/10.1016/j.pnucene.2022.104143. URL: https://www.sciencedirect.com/science/article/pii/S0149197022000233.

[FHZ23] Zicheng Fei, Zhelin Huang, and Xinhai Zhang. “Voltage and temperature information ensembled probabilistic battery health evaluation via deep Gaussian mixture density network”. In: *Journal of Energy Storage* 73 (2023), p. 108587. ISSN: 2352-152X. DOI: https://doi.org/10.1016/j.est.2023.108587. URL: https://www.sciencedirect.com/science/article/pii/S2352152X23019849.

[Kim+23] Gyeongho Kim et al. “A multi-domain mixture density network for tool wear prediction under multiple machining conditions”. In: *International Journal of Production Research* 0.0 (2023), pp. 1–20. DOI: 10.1080/00207543.2023.2289076. URL: https://doi.org/10.1080/00207543.2023.2289076.

[Mit+23] Mihaela Mitici et al. “Prognostics for Lithium-ion batteries for electric Vertical Take-off and Landing aircraft using data-driven machine learning”. In: *Energy and AI* 12 (2023), p. 100233. ISSN: 2666-5468. DOI: https://doi.org/10.1016/j.egyai.2023.100233. URL: https://www.sciencedirect.com/science/article/pii/S2666546823000058.