3. Methodology

In this chapter we will discuss about the dataset that was used, how they were processed in order to be fit to use for training the models. Afterwards the process flow of how the models were built and assessed will be demonstrated.

3.1 Dataset

The data that were used were provided by Hellas-Sat and collected through their telemetry system for Hellas-Sat 3. The rates of all four reaction wheels were provided for a timespan of one year, from 1-1-2022 01:00 UTC to 31-12-2023 23:00 UTC. The measurements that were provided were the hourly average rpm of each wheel, which was calculated by taking the mean of the minimum and the maximum value in a 60 minute window. For each wheel the dataset that was use contained 8749 measurements.

! Figures of the wheel rates before and after the processing to be added.

! Description of plots and any patterns identified, in accordance with the operations schedule that is used by Hellas-Sat.

Additionally, all data were scanned for any missing or corrupted values and an additional step that had to be taken is that we took the absolute values of the measurements. This is related to the fact that the reaction wheels have the ability to rotate both clockwise and counter-clockwise and at all times two of the wheels are spinning CW and two CCW. For purposes of maintenance, once per year the satellite operators make each wheel spin in the opposite direction and therefore we had to filter out of the dataset this transition, in order to avoid confusing the model during the training, because this is not a normal behavior of the wheel that the model needs to be able to predict.

Moreover, during the process of training, experimenting and building the models we came to the conclusion that all values needed to be normalized with respect to the maximum value that the rate of the wheel is allowed to reach, as advised by the constructor and Hellas-Sat. In this way better performance of the models was achieved.

Finally, before the data was fed to the models they had to be split into three subsets: Training, Validation and Testing. For training purposes 70% of the original dataset was used, for validation 20% and for testing 10%. The resulting subsets contained 6124 datapoints for training, 1750 for validation and 802 for testing.

3.2 Model Building

To create a model for timeseries forecasting, there are several parameters that need to be taken into account and combined result into many different architectures that have to be explored.

Those parameters are the number of layers within the model, as well as the type of layers and the number of neurons in each layer. For our case, Long Short Term Memory layers were used. They are a type of Recurrent Neural Network layer and they are widely used for speech recognition, text predictions and more importantly for timeseries forecasting. Another type of layer that was used is the Dense layer, which is the basic type of layer in machine learning.

The following figure displays a straightforward representation of the long short-term memory network. Typically, an LSTM network refers to an RNN with long short-term memory blocks. These blocks are interconnected recurrently and are responsible for maintaining memory. Each memory block contains memory cells that store the state, an input gate for determining what to learn, a forget gate for deciding what to discard, and an output gate for regulating modifications to the content. The critical distinction between LSTM and traditional RNN lies in the inclusion of these gates. The LSTM network utilizes these gates to make informed decisions about preserving relevant information, whereas the traditional RNN merely overwrites its contents at each time step during the modeling process.

A diagram of a mathematical model

Description automatically generated

The standard expression of the LSTM is provided in the following equations:

𝑖𝑡 = 𝜎𝑔(𝑊𝑖𝑠𝑡−1 + 𝑈𝑖𝑥𝑡 + 𝑏𝑖)

𝑜𝑡 = 𝜎𝑔(𝑊𝑜𝑠𝑡−1 + 𝑈𝑜𝑥𝑡 + 𝑏𝑜)

𝑓𝑡 = 𝜎𝑔(𝑊𝑓𝑠𝑡−1 + 𝑈𝑓 𝑥𝑡 + 𝑏𝑓)

̃𝑠𝑡 = 𝜙(𝑊 (𝑜𝑡 ⊙ 𝑠𝑡−1) + 𝑈 𝑥𝑡 + 𝑏)

𝑠𝑡 = 𝑓𝑡 ⊙ 𝑠𝑡−1 + 𝑖

𝑡 ⊙ ̃𝑠𝑡

𝑦𝑡 = 𝑜𝑡 ⊙ 𝜎𝑦(𝑠𝑡)

where 𝑠𝑡 and 𝑠𝑡−1 are the current and previous states; 𝑊 , 𝑈 are the weights in the networks; 𝑏 is the bias; 𝜎𝑔, 𝜙, 𝜎𝑦 are activation functions; and ⊙ denotes the element-wise multiplication.

Other parameters that affect the performance of the model are the learning rate and the optimizer that is used and the input window.

The window size of the input data for an LSTM network refers to the number of time steps or observations that the LSTM considers as a single input sequence. In other words, it determines how much historical information the LSTM can utilize to make predictions or learn patterns in the data. The window size is a crucial hyperparameter in LSTM networks, and selecting an appropriate value depends on the nature of the data and the specific task at hand. A smaller window size may be suitable for tasks with rapid changes, while a larger window size might be more appropriate for tasks with long-term dependencies. In our case, we experimented with window sizes of 24, 48 and 72, which translates to using one, two or three days of data as input, since our data are hourly measurements.

For a more complete exploration of the possible architectures, a tool for hyperparameter optimization was created that utilized Hyperband optimization, so that our research would be more extensive and would include as many candidate architectures as possible.

Hyperband's key idea is to quickly identify and eliminate poorly performing configurations, focusing resources on the most promising ones. This approach allows Hyperband to explore a wide range of hyperparameter configurations more efficiently than traditional grid search.

It starts by sampling a large number of random hyperparameter configurations and evaluates them for a small number of epochs, which is referred to as the "initial" number of epochs. After each epoch, it discards the poorly performing configurations and retains the top-performing configurations. The retained configurations are then allocated more resources (epochs) and trained further.

The process continues, progressively reducing the number of configurations but increasing the number of epochs allocated to each configuration at each round. Eventually, only the best configuration remains and is trained for a large number of epochs.

In summary, Hyperband does not exhaustively test all possible combinations of parameters. Instead, it combines random sampling with early elimination of underperforming configurations to find the best hyperparameter configuration within a specified resource budget.

In Hyperband, the number of configurations tested at each round is determined by a formula:

n = ceil((R / eta^s) \* B), where:

n is the number of configurations at the current round,

R is the maximum number of epochs (max\_epochs) specified for the search,

eta is the factor parameter, and

s is the round number (starting from 0).

The factor parameter in the Hyperband tuner controls the resource allocation and termination rate of the hyperparameter search. It determines the ratio between the number of configurations to evaluate at each round of successive halving. Its purpose is to control the aggressive elimination of poorly performing configurations and the allocation of more resources to promising configurations.

Although the reaction wheels are similar to their characteristics a model for each one had to be created, because they are positioned in different spots within the spacecraft and therefore they experience and are affected by the perturbations in a unique manner each. In the following tables we can see the results of hyperparameter optimization of different architectures for each reaction wheel. As we can see, the results vary among the reaction wheels for similar architectures.

Reaction Wheel 1 architecture results of hyperparameter optimization

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Network | Window Size | Layers | Output Shape | No. of Parameters | Total Parameters | Trainable Parameters |
| RNN | (24, 1) | LSTM | (None, 32) | 7296 | 7329 | 7329 |
|  |  | Dense | (None, 1) | 33 |  |  |
| RNN | (48, 1) | LSTM | (None, 256) | 312320 | 312577 | 312577 |
|  |  | Dense | (None, 1) | 257 |  |  |
| RNN | (48, 1) | LSTM | (None, 64) | 28928 | 45825 | 45825 |
|  |  | Dense | (None, 256) | 16640 |  |  |
|  |  | Dense | (None, 1) | 257 |  |  |
| RNN | (24, 1) | LSTM | (None, 64) | 16896 | 41857 | 41857 |
|  |  | Dense | (None, 128) | 8320 |  |  |
|  |  | Dense | (None, 128) | 16512 |  |  |
|  |  | Dense | (None, 1) | 129 |  |  |
| RNN | (48, 1) | LSTM | (None, None, 128) | 90624 | 119937 | 119937 |
|  |  | LSTM | (None, 32) | 20608 |  |  |
|  |  | Dense | (None, 256) | 8448 |  |  |
|  |  | Dense | (None, 1) | 257 |  |  |
| RNN | (72, 1) | LSTM | (None, None, 192) | 203520 | 273537 | 273537 |
|  |  | LSTM | (None, 64) | 65792 |  |  |
|  |  | Dense | (None, 64) | 4160 |  |  |
|  |  | Dense | (None, 1) | 65 |  |  |
| RNN | (72, 1) | LSTM | (None, None, 64) | 35072 | 245121 | 245121 |
|  |  | LSTM | (None, None, 32) | 12416 |  |  |
|  |  | LSTM | (None, 192) | 172800 |  |  |
|  |  | Dense | (None, 128) | 24704 |  |  |
|  |  | Dense | (None, 1) | 129 |  |  |
| RNN | (72, 1) | LSTM | (None, None, 64) | 35072 | 267585 | 267585 |
|  |  | LSTM | (None, None, 192) | 197376 |  |  |
|  |  | LSTM | (None, 32) | 28800 |  |  |
|  |  | Dense | (None, 64) | 2112 |  |  |
|  |  | Dense | (None, 64) | 4160 |  |  |
|  |  | Dense | (None, 1) | 65 |  |  |

Reaction Wheel 2 architecture results of hyperparameter optimization

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Network | Window Size | Layers | Output Shape | No. of Parameters | Total Parameters | Trainable Parameters |
| RNN | (24, 1) | LSTM | (None, 256) | 287744 | 288001 | 288001 |
|  |  | Dense | (None, 1) | 257 |  |  |
| RNN | (48, 1) | LSTM | (None, 64) | 28928 | 28993 | 28993 |
|  |  | Dense | (None, 1) | 65 |  |  |
| RNN | (48, 1) | LSTM | (None, 128) | 90624 | 94785 | 94785 |
|  |  | Dense | (None, 32) | 4128 |  |  |
|  |  | Dense | (None, 1) | 33 |  |  |
| RNN | (24, 1) | LTSM | (None, 64) | 16896 | 50049 | 50049 |
|  |  | Dense | (None, 256) | 16640 |  |  |
|  |  | Dense | (None, 64) | 16448 |  |  |
|  |  | Dense | (None, 1) | 65 |  |  |
| RNN | (48, 1) | LTSM | (None, None, 192) | 185088 | 382721 | 382721 |
|  |  | LTSM | (None, 128) | 164352 |  |  |
|  |  | Dense | (None, 256) | 33027 |  |  |
|  |  | Dense | (None, 1) | 257 |  |  |
| RNN | (72, 1) | LTSM | (None, None , 192) | 203520 | 233409 | 233409 |
|  |  | LTSM | (None, 32) | 28800 |  |  |
|  |  | Dense | (None, 32) | 1056 |  |  |
|  |  | Dense | (None, 1) | 33 |  |  |
| RNN | (72, 1) | LSTM | (None, None, 32) | 13440 | 217217 | 217217 |
|  |  | LSTM | (None, None, 192) | 172800 |  |  |
|  |  | LSTM | (None, 32) | 28800 |  |  |
|  |  | Dense | (None, 64) | 2112 |  |  |
|  |  | Dense | (None, 1) | 65 |  |  |
| RNN | (72, 1) | LSTM | (None, None, 128) | 102912 | 538625 | 538625 |
|  |  | LSTM | (None, None, 192) | 246528 |  |  |
|  |  | LSTM | (None, 128) | 164352 |  |  |
|  |  | Dense | (None, 128) | 16512 |  |  |
|  |  | Dense | (None, 64) | 8256 |  |  |
|  |  | Dense | (None, 1) | 65 |  |  |

Reaction Wheel 3 architecture results of hyperparameter optimization

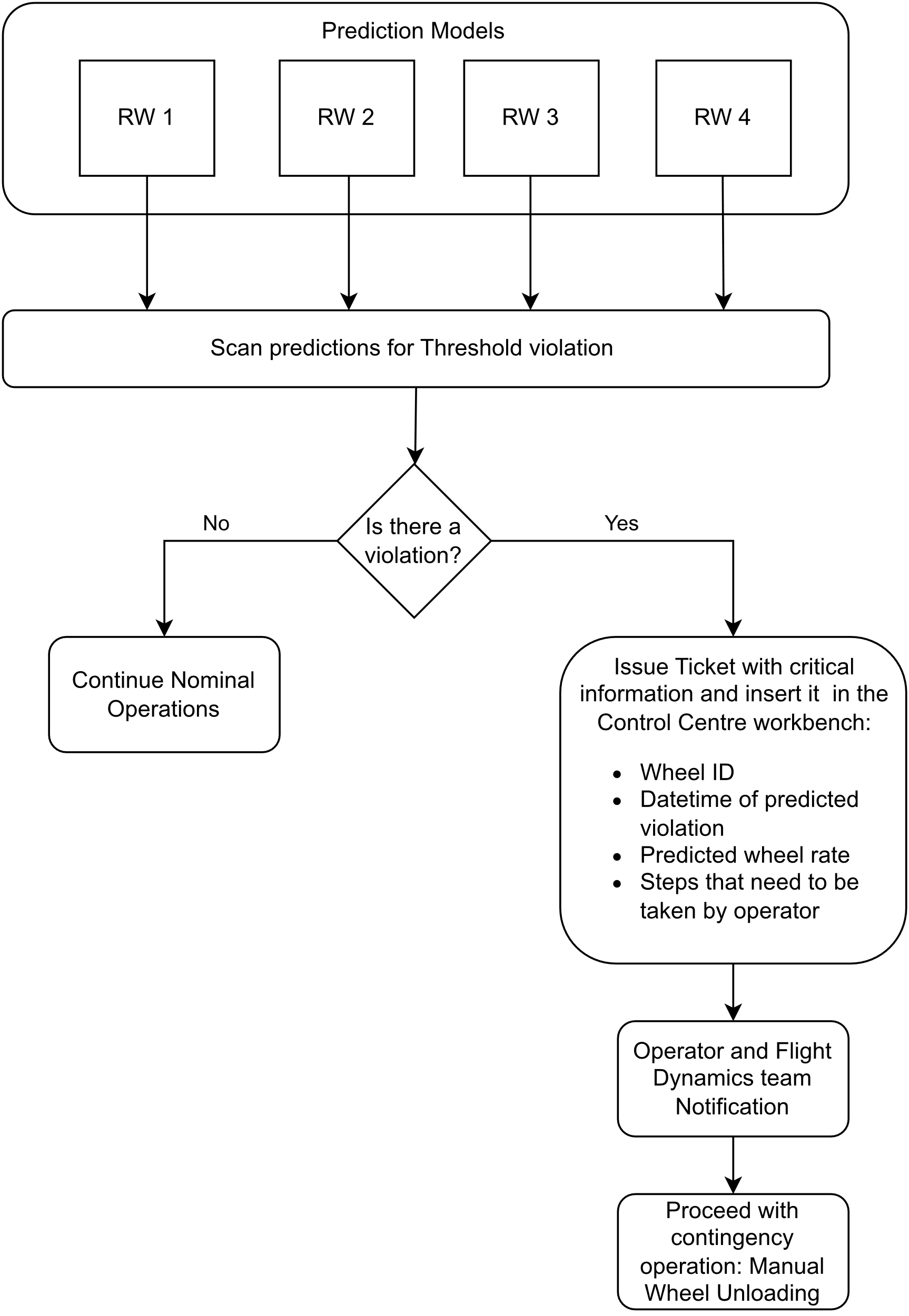
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Network | Window Size | Layers | Output Shape | No. of Parameters | Total Parameters | Trainable Parameters |
| RNN | (24, 1) | LSTM | (None, 32) | 7296 | 7329 | 7329 |
|  |  | Dense | (None, 1) | 33 |  |  |
| RNN | (48, 1) | LSTM | (None, 256) | 312320 | 312577 | 312577 |
|  |  | Dense | (None, 1) | 257 |  |  |
| RNN | (48, 1) | LSTM | (None, 128) | 90624 | 107265 | 107265 |
|  |  | Dense | (None, 128) | 16512 |  |  |
|  |  | Dense | (None, 1) | 129 |  |  |
| RNN | (24, 1) | LSTM | (None, 128) | 66560 | 149121 | 149121 |
|  |  | Dense | (None, 256) | 33024 |  |  |
|  |  | Dense | (None, 192) | 49344 |  |  |
|  |  | Dense | (None, 1) | 193 |  |  |
| RNN | (48, 1) | LSTM | (None, None, 256) | 312320 | 706817 | 706817 |
|  |  | LSTM | (None, 192) | 344832 |  |  |
|  |  | Dense | (None, 256) | 49408 |  |  |
|  |  | Dense | (None, 1) | 257 |  |  |
| RNN | (72, 1) | LSTM | (None, None, 192) | 203520 | 536449 | 536449 |
|  |  | LSTM | (None, 192) | 295680 |  |  |
|  |  | Dense | (None, 192) | 37056 |  |  |
|  |  | Dense | (None, 1) | 193 |  |  |
| RNN | (72, 1) | LSTM | (None, None, 192) | 203520 | 417537 | 417537 |
|  |  | LSTM | (None, None, 32) | 28800 |  |  |
|  |  | LSTM | (None, 192) | 172800 |  |  |
|  |  | Dense | (None, 64) | 12352 |  |  |
|  |  | Dense | (None, 1) | 65 |  |  |
| RNN | (72, 1) | LSTM | (None, None, 64) | 35072 | 353825 | 353825 |
|  |  | LSTM | (None, None, 32) | 12416 |  |  |
|  |  | LSTM | (None, 256) | 295936 |  |  |
|  |  | Dense | (None, 32) | 8224 |  |  |
|  |  | Dense | (None, 64) | 2112 |  |  |
|  |  | Dense | (None, 1) | 65 |  |  |

Reaction Wheel 4 architecture results of hyperparameter optimization

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Network | Window Size | Layers | Output Shape | No. of Parameters | Total Parameters | Trainable Parameters |
| RNN | (24, 1) | LSTM | (None, 192) | 166656 | 166849 | 166849 |
|  |  | Dense | (None, 1) | 193 |  |  |
| RNN | (48, 1) | LSTM | (None, 32) | 10368 | 10401 | 10401 |
|  |  | Dense | (None, 1) | 33 |  |  |
| RNN | (48, 1) | LSTM | (None, 256) | 312320 | 328833 | 328833 |
|  |  | Dense | (None, 64) | 16448 |  |  |
|  |  | Dense | (None, 1) | 65 |  |  |
| RNN | (24, 1) | LSTM | (None, 128) | 66560 | 79393 | 79393 |
|  |  | Dense | (None, 32) | 4128 |  |  |
|  |  | Dense | (None, 256) | 8448 |  |  |
|  |  | Dense | (None,1) | 257 |  |  |
| RNN | (48, 1) | LSTM | (None, None, 128) | 90624 | 148481 | 148481 |
|  |  | LSTM | (None, 64) | 49408 |  |  |
|  |  | Dense | (None, 128) | 8320 |  |  |
|  |  | Dense | (None, 1) | 129 |  |  |
| RNN | (72, 1) | LSTM | (None, None, 192) | 203520 | 281985 | 281985 |
|  |  | LSTM | (None, 64) | 65792 |  |  |
|  |  | Dense | (None, 192) | 12480 |  |  |
|  |  | Dense | (None, 1) | 193 |  |  |
| RNN | (72, 1) | LSTM | (None, None, 256) | 336896 | 764289 | 764289 |
|  |  | LSTM | (None, None, 64) | 82176 |  |  |
|  |  | LSTM | (None, 256) | 328704 |  |  |
|  |  | Dense | (None, 64) | 16448 |  |  |
|  |  | Dense | (None, 1) | 65 |  |  |
| RNN | (72, 1) | LSTM | (None, None, 192) | 203520 | 396865 | 396865 |
|  |  | LSTM | (None, None, 128) | 164352 |  |  |
|  |  | LSTM | (None, 32) | 20608 |  |  |
|  |  | Dense | (None, 128) | 4224 |  |  |
|  |  | Dense | (None, 32) | 4128 |  |  |
|  |  | Dense | (None, 1) | 33 |  |  |

3.3 Ticket Creation

Once the models for each reaction wheel are built and trained, we can make predictions for each one, to detect if their rate is going to exceed a specific threshold and then create a ticket in the Spacecraft Control Centre with the necessary information, so that the engineers, operators and flight dynamics team can act accordingly. The following flow diagram explains the procedure in which the ticket will be created and the steps that need to take place.



Firstly, the predictions are created for each Reaction Wheel for a certain period. Then the results need to be examined, in order to assess if any of the predictions exceed a set threshold.

If none of the wheels exceed the set limit, then no action needs to be taken and operations can proceed nominally. However, if the predictions indicate that one of the wheels will surpass the limit at a specific time, a ticket needs to be created in the operation center workbench, so that the operators can be notified that a manual wheel unloading will need to take place.

To achieve that a process of scanning the prediction results was created, which identifies which wheel will exceed the limit and when, and then creates a ticket with the appropriate information on the workbench, by creating an entry in the right tables in its database. The information that is input is the date of the procedure, which is the date that the rate will pass the threshold, the predicted rate of the wheel, which wheel will it be and the steps of the procedure that will need to be followed.

The ticket shall be marked as unverified, so that the flight dynamics team can process the information that is given and decide in what direction the unloading will be performed and if there is a more efficient time to execute the procedure than the one that is predicted.

! Screenshots of the tables, or ticket to be added.