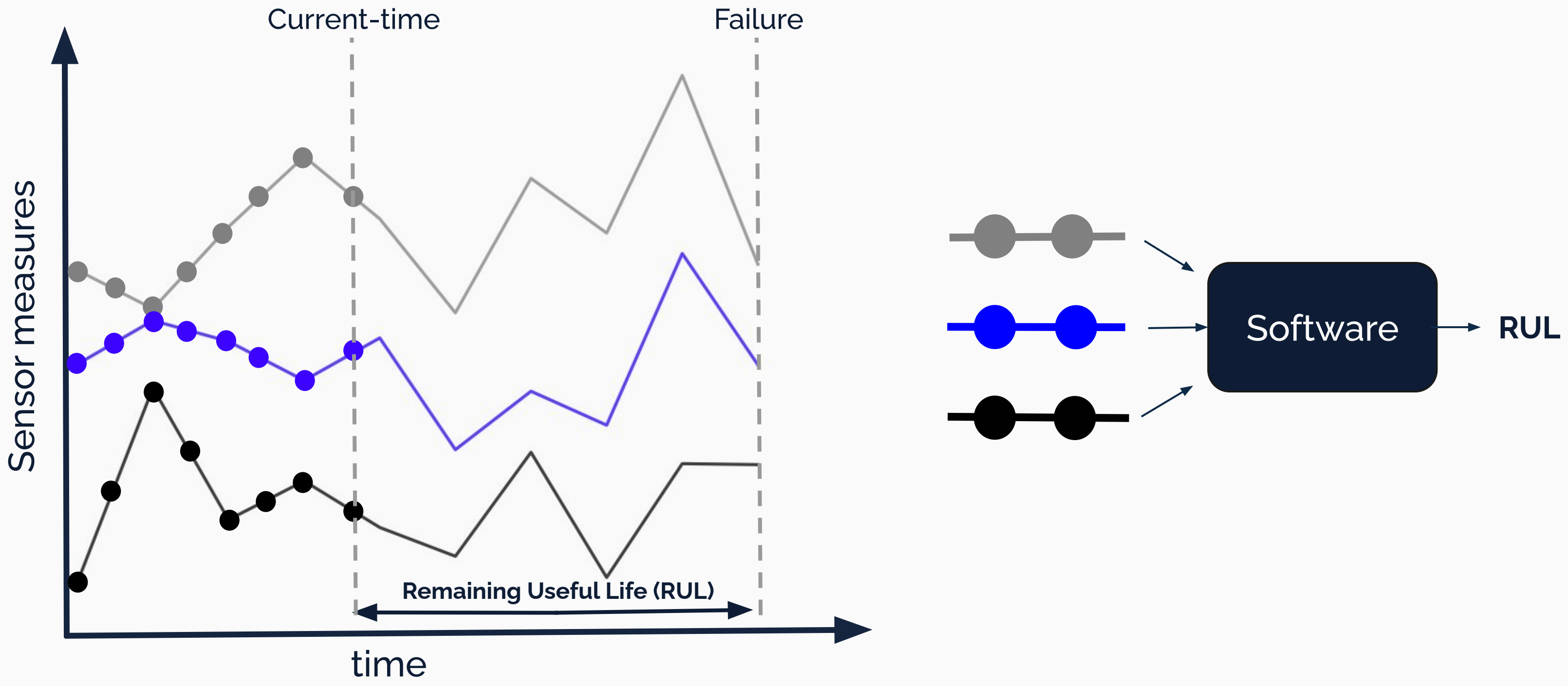
TIME-TO-FAILURE PREDICTION (TTF)

Forecasting plane engine failure with sensors data

Lorenzo Suppa, Vito Perrucci and Tanguy Dugas du Villard

Goal: Predicting airplane's engine failure before it happens



Objectives and Value Proposition:

Offer continuous maintenance assistance

Decrease components production

Decrease maintenance time and costs

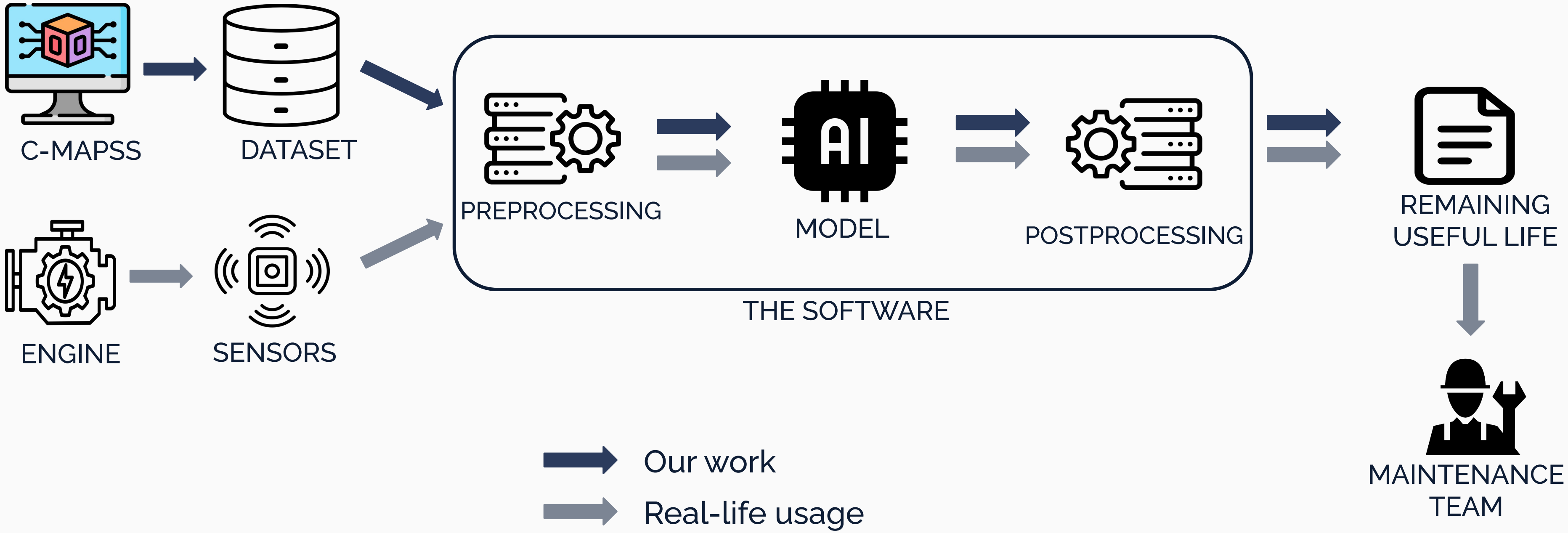
9 INDUSTRY, INNOVATION AND INFRASTRUCTURE

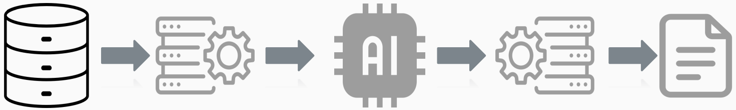
12 RESPONSIBLE CONSUMPTION AND PRODUCTION

8 DECENT WORK AND ECONOMIC GROWTH

SUSTAINABLE DEVELOPMENT GOALS

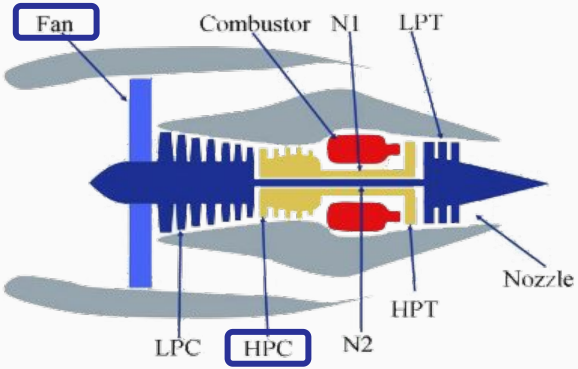
Functional diagram:





Challenges of the data

How to deal with different types of failure?

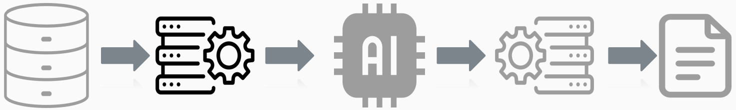


How to adapt forecasts to environment settings?

Air temperature

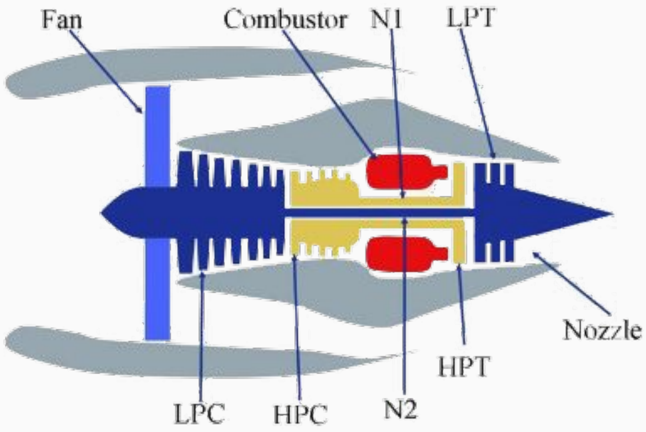
Altitude

Speed



Research questions

- Which are the most relevant features to collect ?

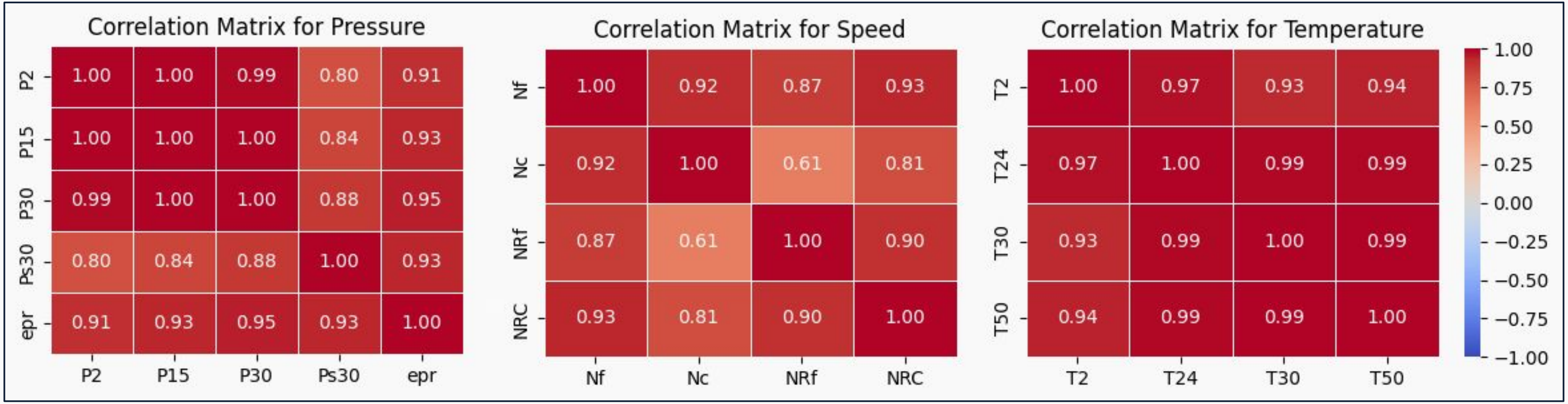


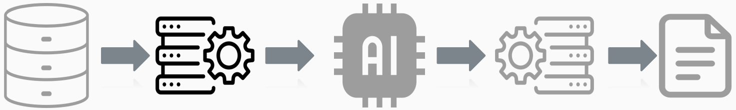
Symbol	*Description*	*Units*
T2	Total temperature at fan inlet	°R
T24	Total temperature at LPC outlet	°R
T30	Total temperature at HPC outlet	°R
T50	Total temperature at LPT outlet	°R
P2	Pressure at fan inlet	psia
P15	Total pressure in bypass-duct	psia
P30	Total pressure at HPC outlet	psia
Nf	Physical fan speed	rpm
Nc	Physical core speed	rpm
epr	Engine pressure ratio (P50/P2)	--
Ps30	Static pressure at HPC outlet	psia
phi	Ratio of fuel flow to Ps30	pps/psi
NRf	Corrected fan speed	rpm
NRc	Corrected core speed	rpm
BPR	Bypass Ratio	--
farB	Burner fuel-air ratio	--
htBleed	Bleed Enthalpy	--
Nf_dmd	Demanded fan speed	rpm
PCNfR_dmd	Demanded corrected fan speed	rpm
W31	HPT coolant bleed	lbm/s
W32	LPT coolant bleed	lbm/s

Research questions

- Which are the most relevant features to collect ?

➔ Correlation Matrices





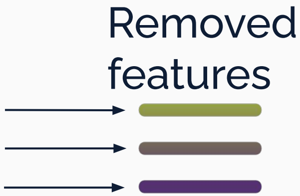
Research questions

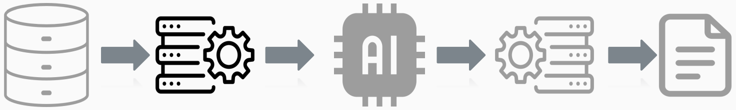
- Which are the most relevant features to collect ?

➔ Backward Elimination

Backward Elimination

1. Linear regression
$$\hat{\beta} = \operatorname{argmin} (X^T\beta - y)^2$$
2. Hypothesis tests: $\hat{\beta}_i = 0$
3. p-values of the test: p_i
4. Worst feature: $i_0 = \operatorname{argmin} p_i$
5. If p_{i_0} > 0.1: Remove i_0 from X and go back to 1.
Else: stop the algorithm





Research questions

- Which are the most relevant features to collect ?

Highly correlated temperatures

Highly correlated pressures

Highly correlated speeds

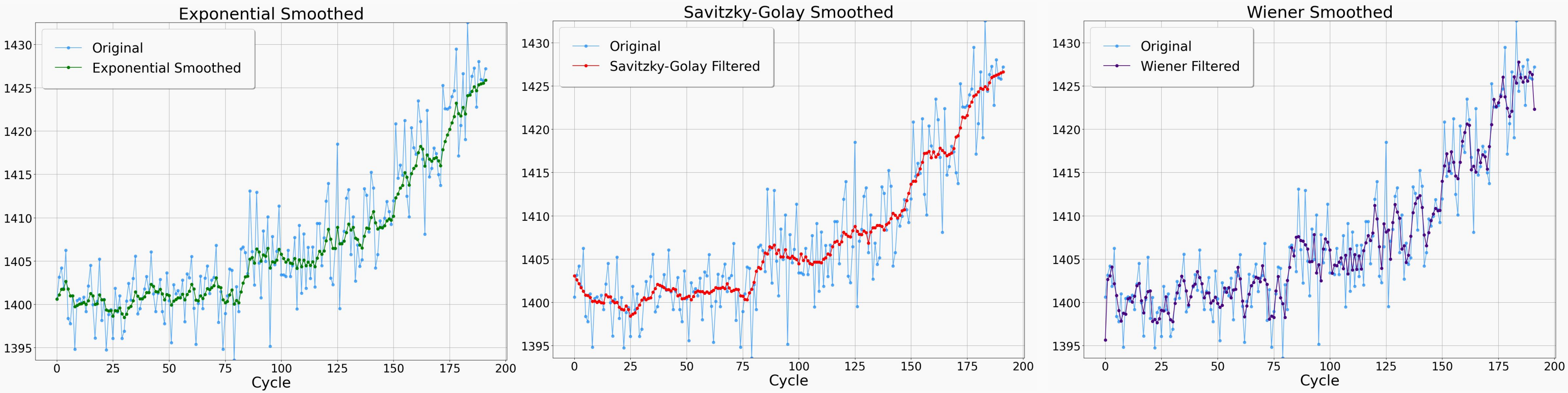
Not enough informative

21 features => 13 features

Symbol	*Description*	*Units*
T2	Total temperature at fan inlet	°R
~~**T24**~~	~~Total temperature at LPC outlet~~	~~°R~~
T30	Total temperature at HPC outlet	°R
~~**T50**~~	~~Total temperature at LPT outlet~~	~~°R~~
P2	Pressure at fan inlet	psia
~~**P15**~~	~~Total pressure in bypass-duct~~	~~psia~~
~~**P30**~~	~~Total pressure at HPC outlet~~	~~psia~~
~~**Nf**~~	~~Physical fan speed~~	~~rpm~~
Nc	Physical core speed	rpm
~~**epr**~~	~~Engine pressure ratio (P50/P2)~~	
Ps30	Static pressure at HPC outlet	psia
phi	Ratio of fuel flow to Ps30	pps/psi
NRf	Corrected fan speed	rpm
~~**NRc**~~	~~Corrected core speed~~	~~rpm~~
BPR	Bypass Ratio	--
~~**farB**~~	~~Burner fuel-air ratio~~	~~--~~
htBleed	Bleed Enthalpy	--
Nf_dmd	Demanded fan speed	rpm
PCNfR_dmd	Demanded corrected fan speed	rpm
W31	HPT coolant bleed	lbm/s
W32	LPT coolant bleed	lbm/s

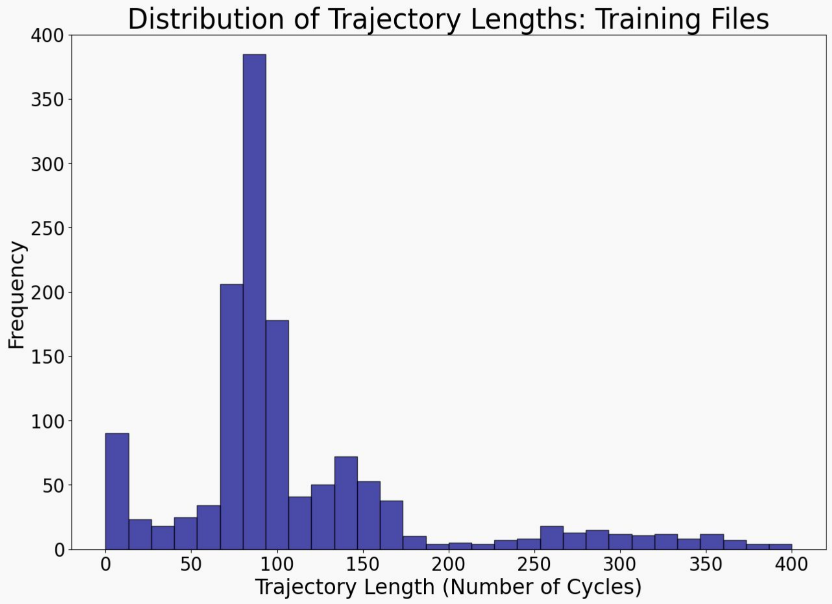
Research questions

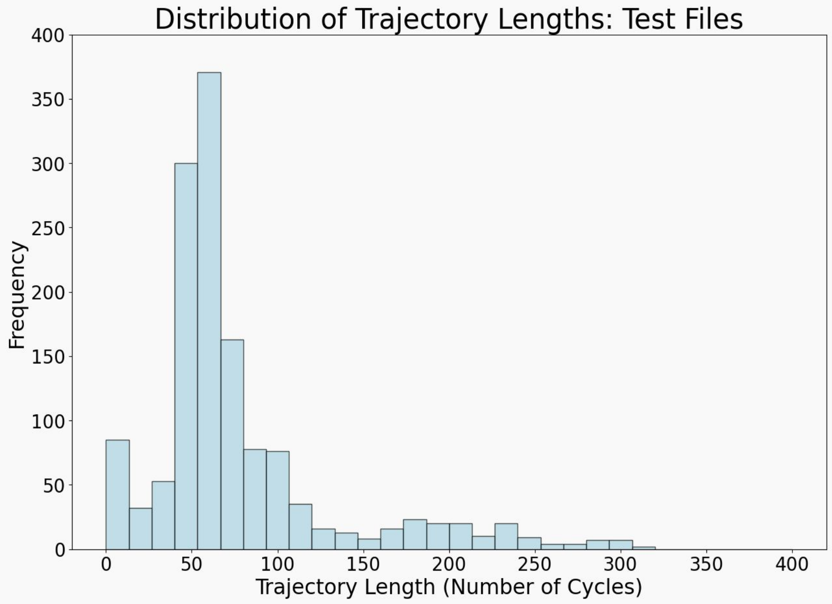
- How can sensor noise in time-series data be effectively mitigated to improve the accuracy in our RUL predictions?



Research questions

- How can we deal with trajectories having different length ?

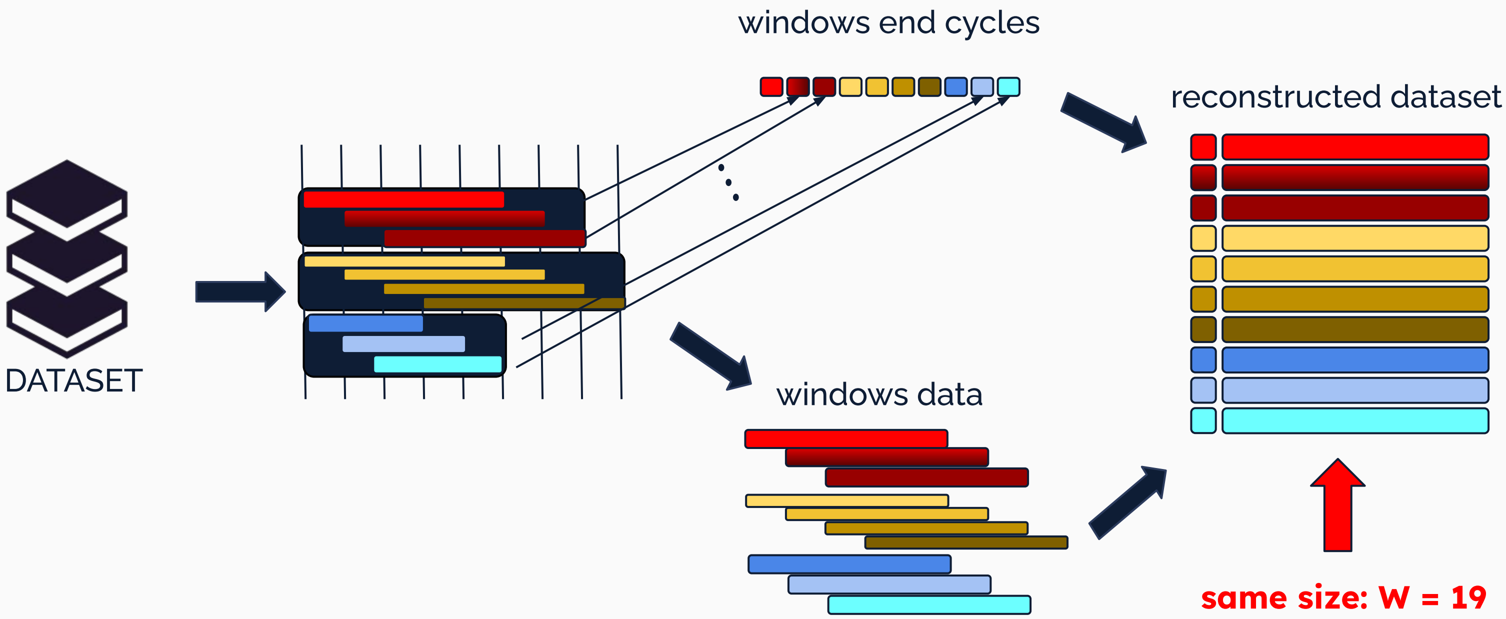




Research questions

- How can we deal with trajectories having different length ?

→ Windows



EncoderRegressor

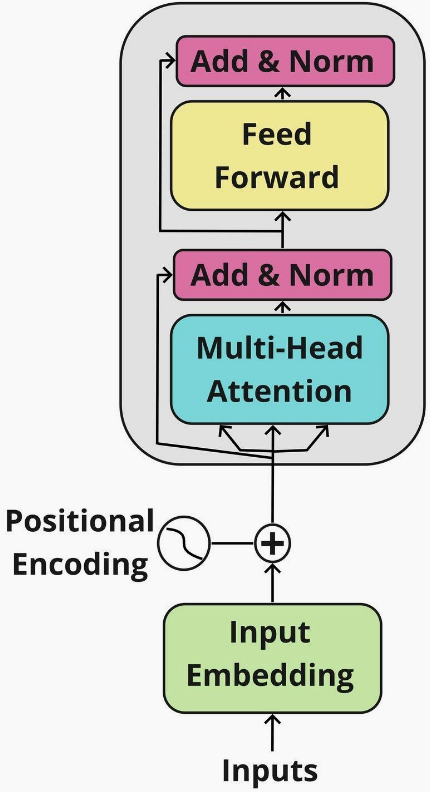
Key changes and chosen hyperparameters:

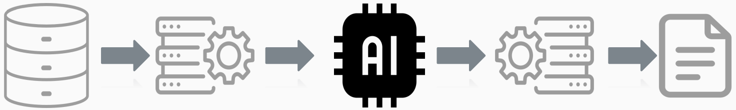
- **Embedding & Positional Encoding**:
  – Positional encoding can be learnable.

- **Within Transformer Encoder Layers**:
  – Batch normalization over Layer normalization to effectively manage possible outliers

- **Output Head**:
  – Pre-training: Maps high-dimensional representations back to original input dimension for reconstruction.
  – Regression: Applies mean pooling followed by a fully-connected head for single RUL prediction.

- **Hyperparameters**:
  – Batch size=128, embedding_dim=256, num_heads=16, num_Encoder_layers=4





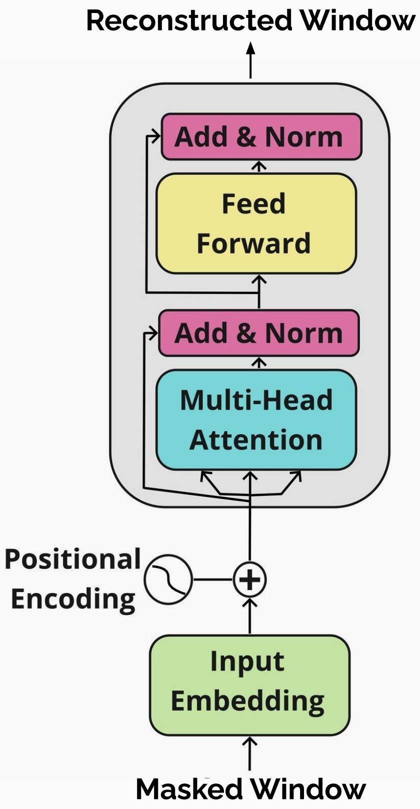
2-Phase Training: Pre-Training

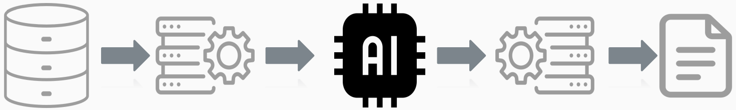
1. Self-supervised input reconstruction

– Randomly mask approximately 15% of window's values.
Another layer at the end of the flow is added to maps the high-dimensional representations back to the original input dimension

– Objective: Reconstructing the original values at the masked position using MSE loss.

– Our intention: Make the model learn contextual and temporal dependencies with few training epochs.





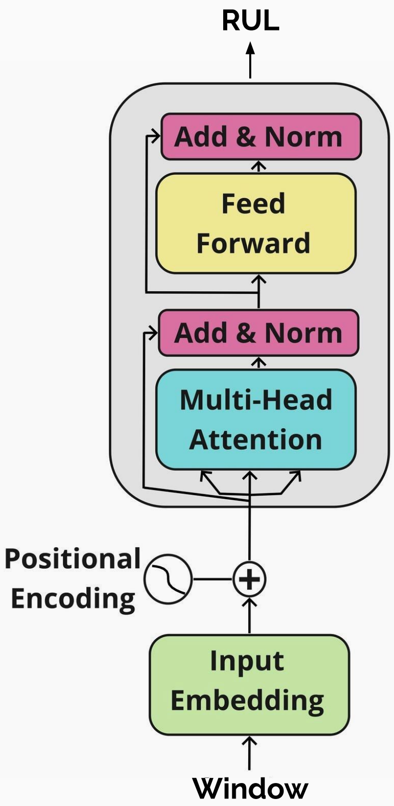
2-Phase Training: Fine-Tuning

2. Self-supervised RUL prediction

– Transfer pre-trained weights to a regression model with a new head that aggregates the sequence via mean pooling and maps the representation to a single RUL prediction.

– Objective: Fine-tune the model on the regression task predicting RUL for each window.

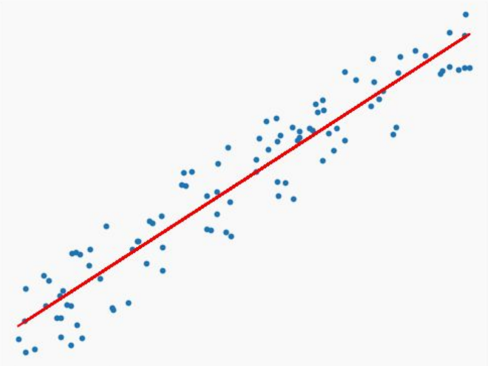
– Our intention: Leverage the robust features learned during pre-training to efficiently adapt the model for precise RUL estimation with minimal additional training epochs.



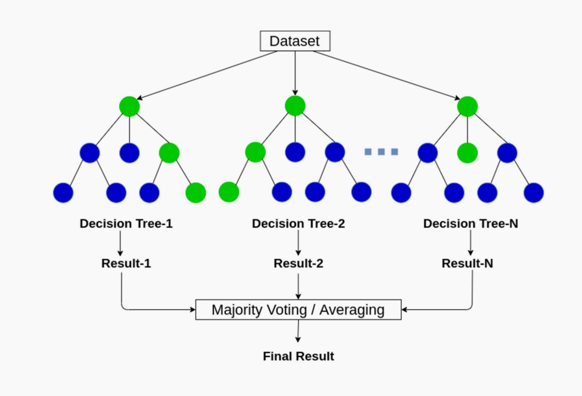
Baseline models

Other models have been trained on the same dataset, with the same preprocessing steps, in order to compare their performance with our EncoderRegressor.
Results and comparisons will be further discussed.

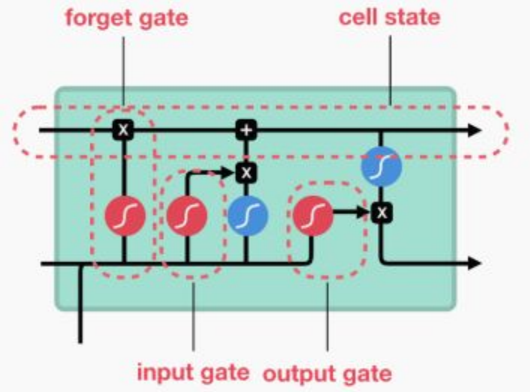
Linear regression



Random forest



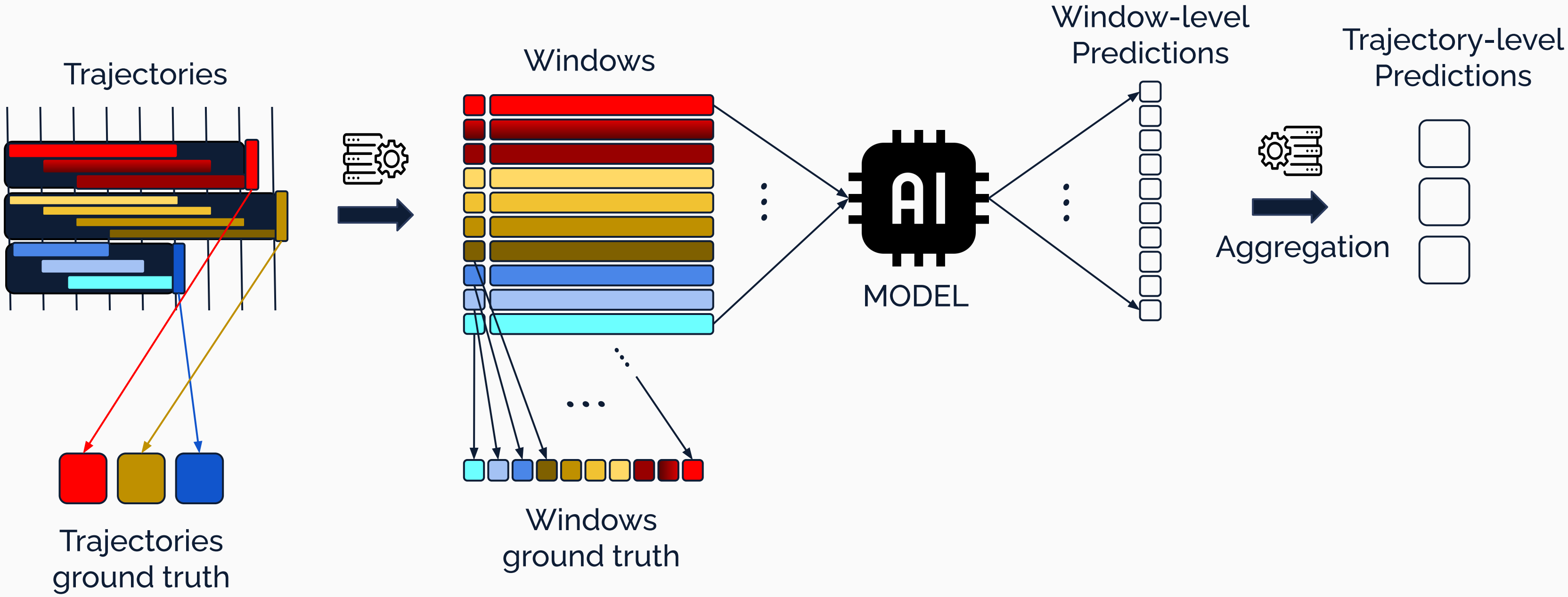
Long Short-Term Memory



Research questions

- How can we deal with trajectories having different length?

→ Fixed window size creation

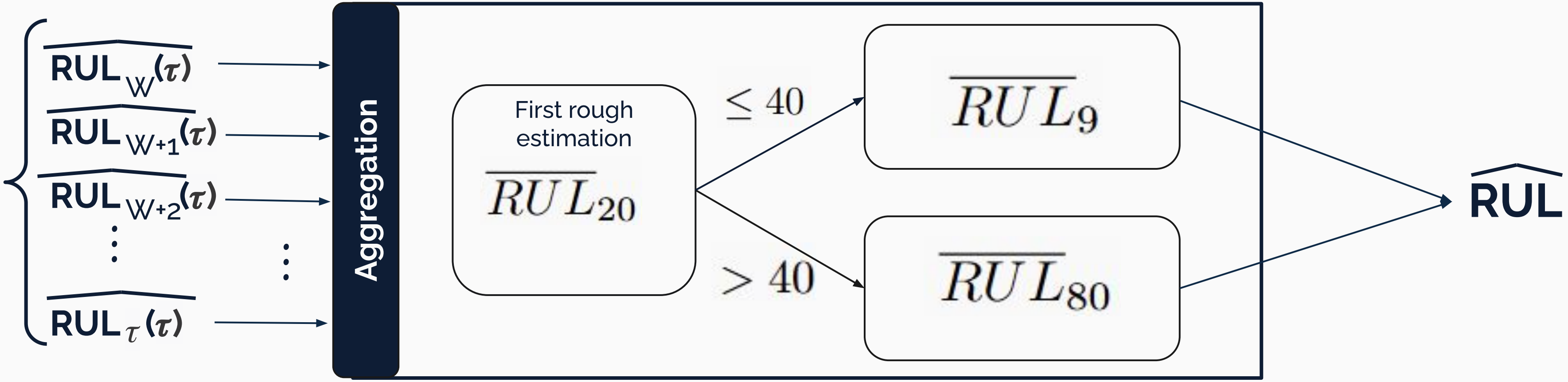


Research questions

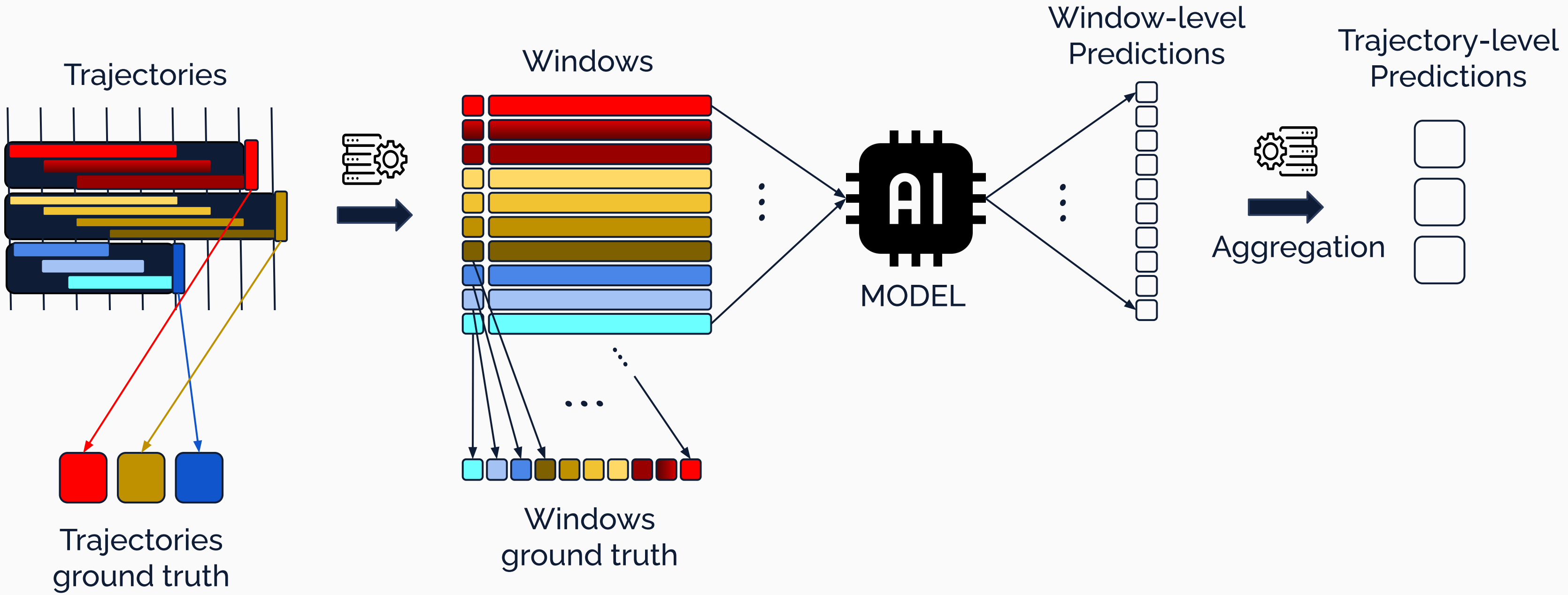
- How can we deal with trajectories having different length ?

➔ Aggregation: Unweighted average over the last predictions

$$\overline{RUL}_N = \frac{1}{N} \sum_{t=\tau-N+1}^{\tau} \widehat{RUL}_t(\tau)$$

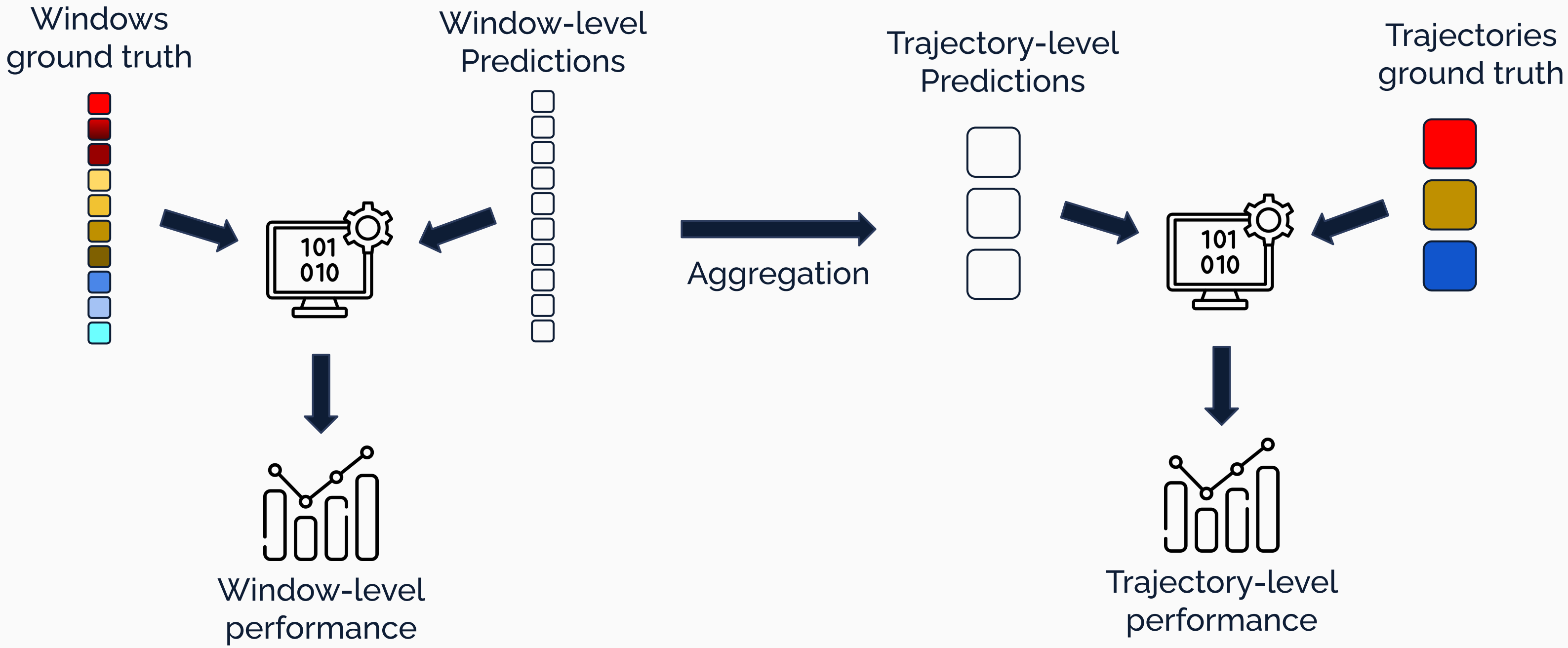


Results



Trajectories
Windows
Window-level
Predictions
Trajectory-level
Predictions
MODEL
Aggregation
Trajectories
ground truth
Windows
ground truth

Results



Windows
ground truth
Window-level
Predictions
101
010
Window-level
performance
Aggregation
Trajectory-level
Predictions
Trajectories
ground truth
101
010
Trajectory-level
performance

Window-Level Performance Analysis

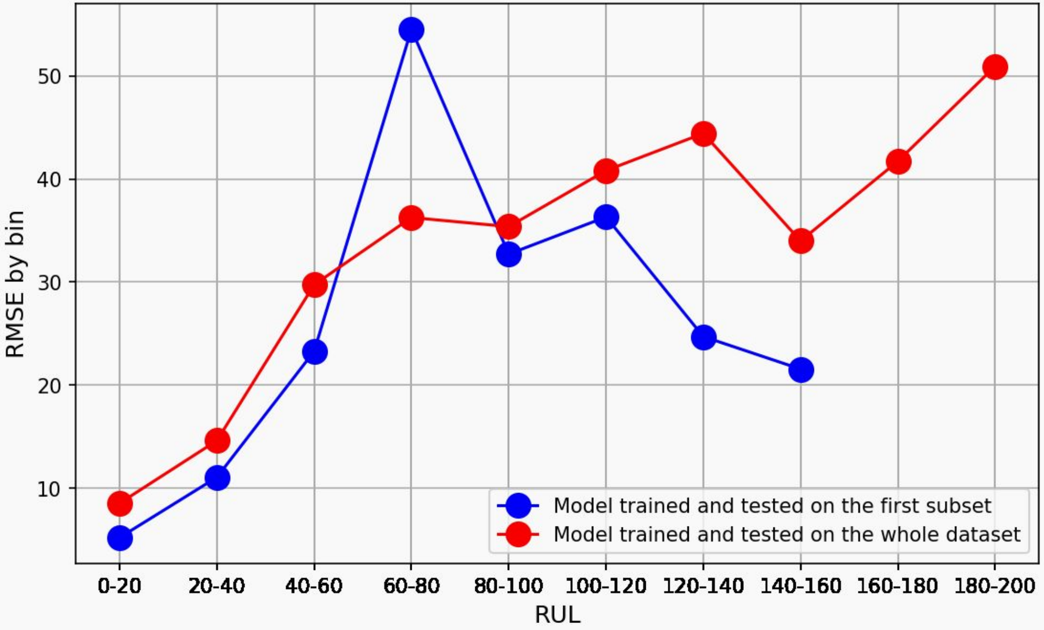
the EncoderRegressor exhibits excellent window-level performance, consistently achieving lower RMSE values in low RUL scenarios crucial for early failure detection

Some metrics on RMSE:

Global RMSE : 41.47
<50 RUL RMSE: 18.15
<20 RUL RMSE: 5.05

Trajectory-Level Performance Analysis

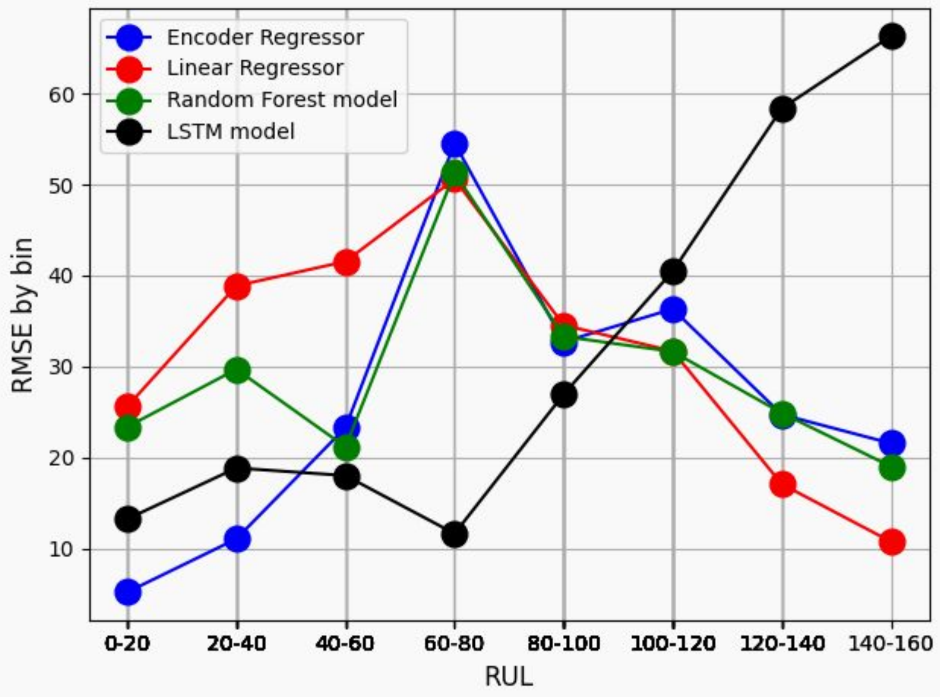
Aggregated forecasts reveal that while the model excels at predicting imminent failures with low RMSE, its error increases gradually as the RUL



On first subset
(1 failure type, 1 flight condition)
Global RMSE : 30.70
<20 RUL RMSE: 5.05

On whole dataset :
(2 failure types, 6 flight conditions)
Global RMSE : 35.20
<20 RUL RMSE: 8.56

Comparison against Other Models



Encoder Regressor :
Global RMSE : 30.70
<20 RUL RMSE: 5.05

Linear Regressor :
Global RMSE : 33.92
<20 RUL RMSE: 25.59

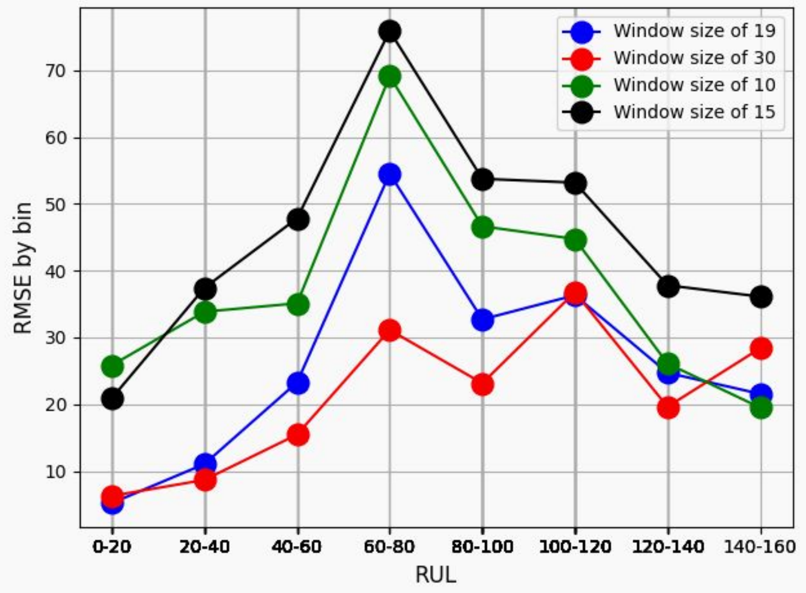
Random Forest model:
Global RMSE : 32.47
<20 RUL RMSE: 23.36

LSTM model:
Global RMSE : 32.84
<20 RUL RMSE: 13.23

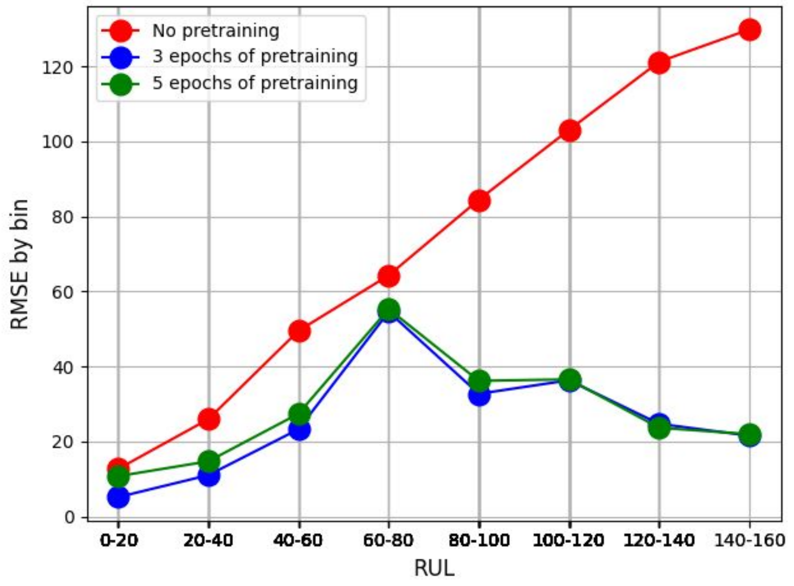
Impact of Window Size and Pretraining

Increasing the window size and the number of pretraining epochs appears to improve performance. However, excessive increases may lead to overfitting.

Windows size analysis

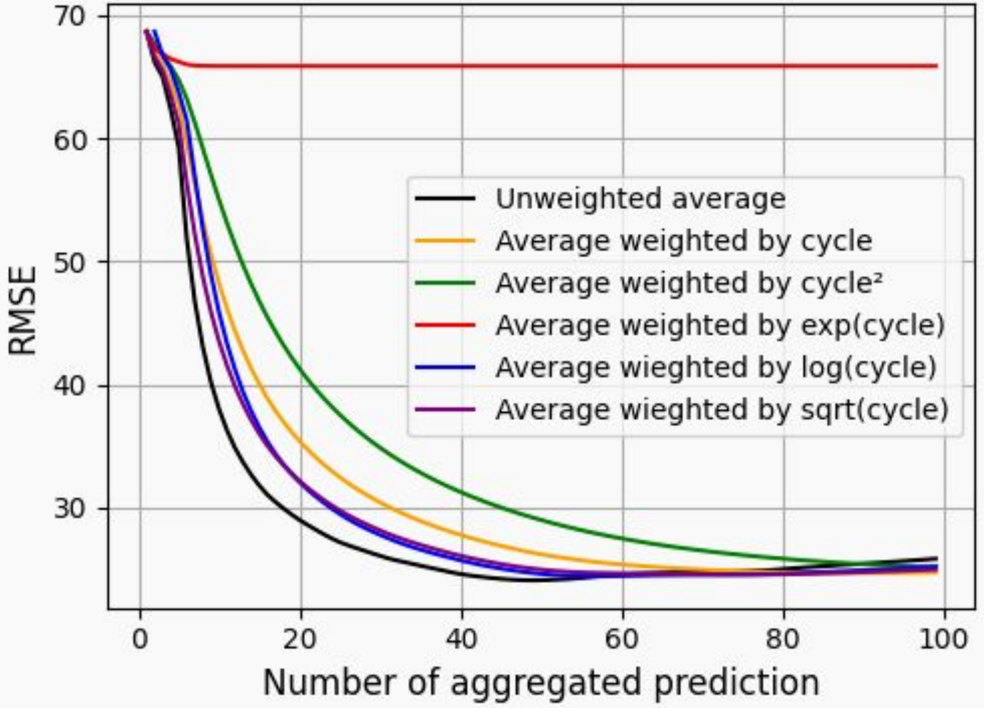


Pretraining analysis



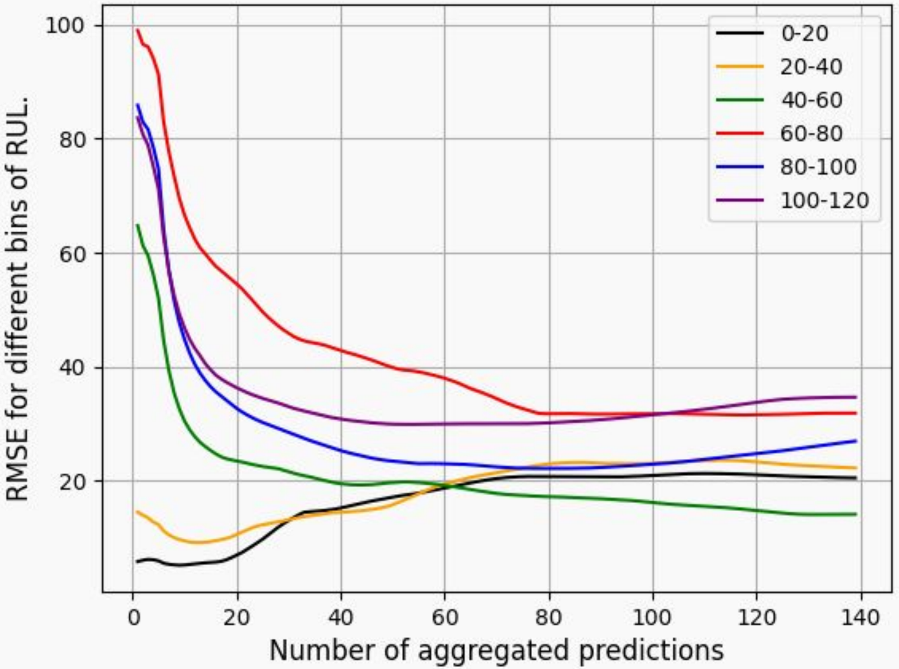
Impact of Aggregation Strategies

Splitting the trajectories by short-term and long-term failure cases, the optimal number of windows is dependent on the time remaining until failure



Number of Aggregated Windows

After selecting the unweighted average as the aggregation strategy, we analyze the impact of the number of aggregated windows on the final prediction.

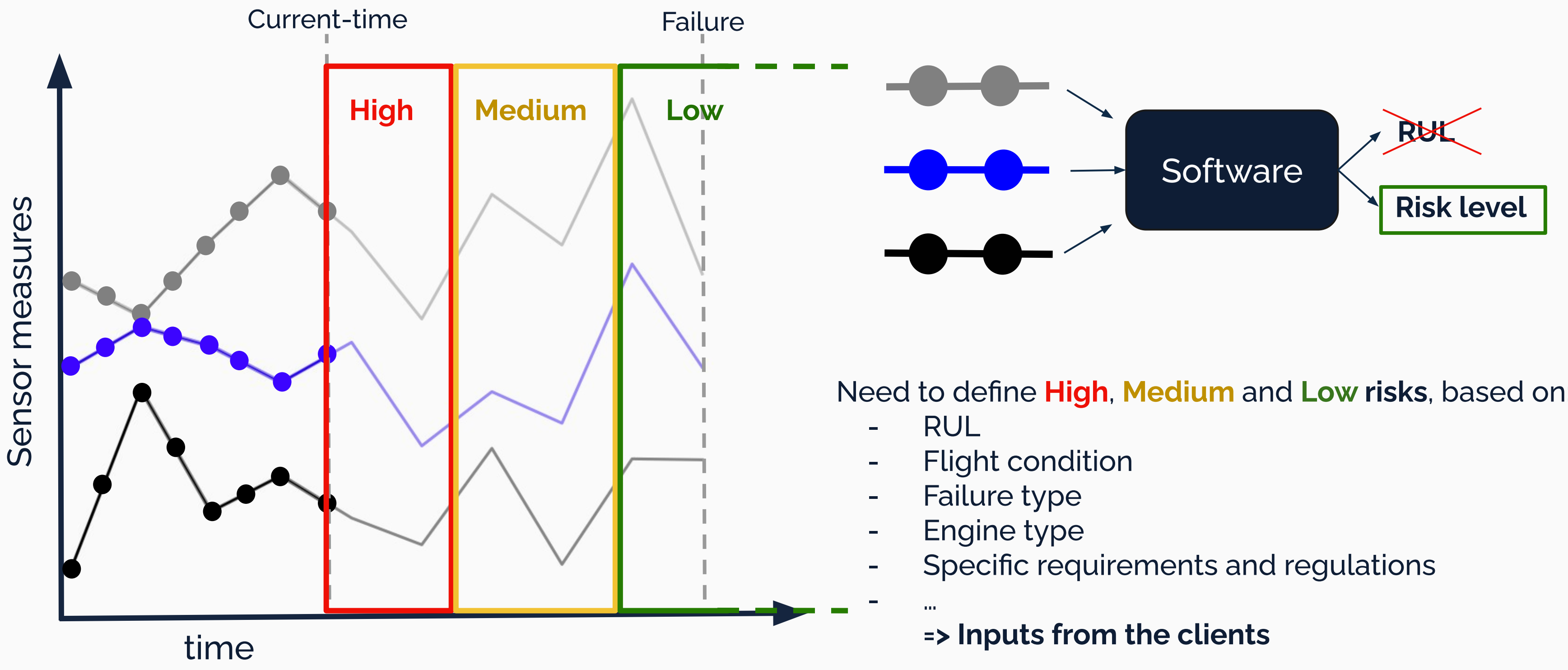


Unweighted Average Aggregation

On Trajectories with RUL < 40:
Best Number of predictions: 9

On Trajectories with RUL > 40:
Best Number of predictions: 80

Further extension: Classifying risk of failure



Need to define **High**, **Medium** and **Low** **risks**, based on:

- RUL
- Flight condition
- Failure type
- Engine type
- Specific requirements and regulations
- ...

=> Inputs from the clients

Any question ?

- Tanguy Dugas du Villard
- Vito Perrucci
- Lorenzo Suppa