PHM Society 2025 Data Challenge

DIT Semestral Project

Ondřej Baštař

Jan 4, 2025

Some extra info

Agenda

1. Problem Description
2. My Approach

Problem Description

Problem statement

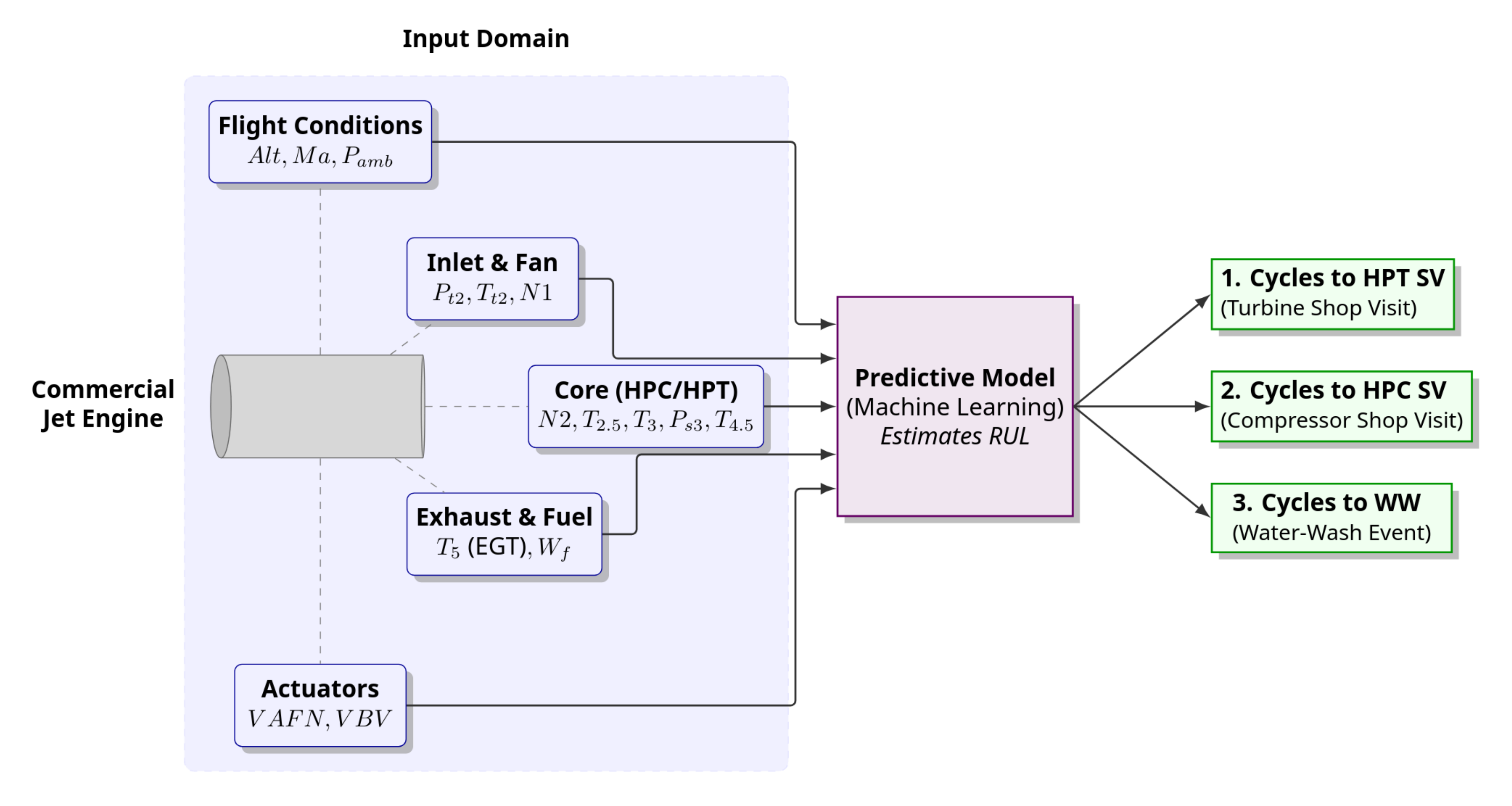

Figure 1: Problem diagram

Dataset Specifications

Sensor Data (16 Features)

- Flight: Alt, Ma, P_{amb}
- Temps: $T_{t_2}, T_{2.5}, T_3, T_{4.5}, T_5$
- Pressures: $P_{t_2}, P_{s_3}, P_{2.5}$
- Rotors: N_1 (Fan), N_2 (Core)
- Actuators: VAFN, VBV

Prediction Targets (RUL)

1. HPT Shop Visit
   (High Pressure Turbine)
2. HPC Shop Visit
   (High Pressure Compressor)
3. Water Wash
   (Routine Maintenance)

Dataset Specifications

Rough amounts of data:

- Mesaurement from 4 engines
- ~2000 datapoints per engine at 8 flight states (landed, flying, takeoff, …)
- Events measured per engine:
  - WW (Water Wash): ~20 per engine
  - HPC Shop Visit: ~3 per engine
  - HPT Shop Visit: ~6 per engine

In conclusion the data are limited for advanced machine learning methods.

Model Evaluation

- Competition scoring function

$$W(y, \hat{y}) = \begin{cases} \frac{2}{1+0.02 \cdot y} & \text{if } \hat{y} \geq y \quad \text{(Late prediction)} \\ \frac{1}{1+0.02 \cdot y} & \text{if } \hat{y} < y \quad \text{(Early prediction)} \end{cases}$$

$$S(y, \hat{y}) = W(y, \hat{y}) \cdot (\hat{y} - y)^2$$

My Approach

Baseline Model

- Simple baseline model for comparison.
- Based on XGBoost (Gradient Boosting Trees)
- Mean score on 4 engines is 93.60

Engine SN	WW	HPC	HPT	Final
101	38.17	84.66	136.81	86.55
102	50.56	203.16	78.84	110.85
103	42.66	159.65	42.89	81.73
104	53.60	167.37	64.82	95.26
Mean	46.25	153.71	80.84	93.60
Std	7.09	49.78	40.14	12.79

Kalman Filter for health tracking

1. Trained a simple model for healthy engine operation such that:
    - $f(\text{flight state}) = \left(\hat{T}_1, \hat{T}_2, \ldots\right)^T = \hat{\boldsymbol{z}}_k$
2. Calculate residuals:
    - $\boldsymbol{r}_k = \boldsymbol{z}_k - \hat{\boldsymbol{z}}_k$
    - $\boldsymbol{z}_k$ is vector of sensor values at time k.
    - size of elements of vector $\boldsymbol{r}_k$ correlates with engine health.
3. Kalman filter for tracking health index for each sesnor residual.
    - Using N simple constant velocity models with each element of of $\boldsymbol{r}_k$ as outputs.

$$\boldsymbol{x}_k = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} \text{Health (residual)} \\ \text{Degradation rate} \end{pmatrix} \quad A = \begin{pmatrix} 1 & \Delta t \\ 0 & 1 \end{pmatrix}$$

Visualisation of the filtered health state

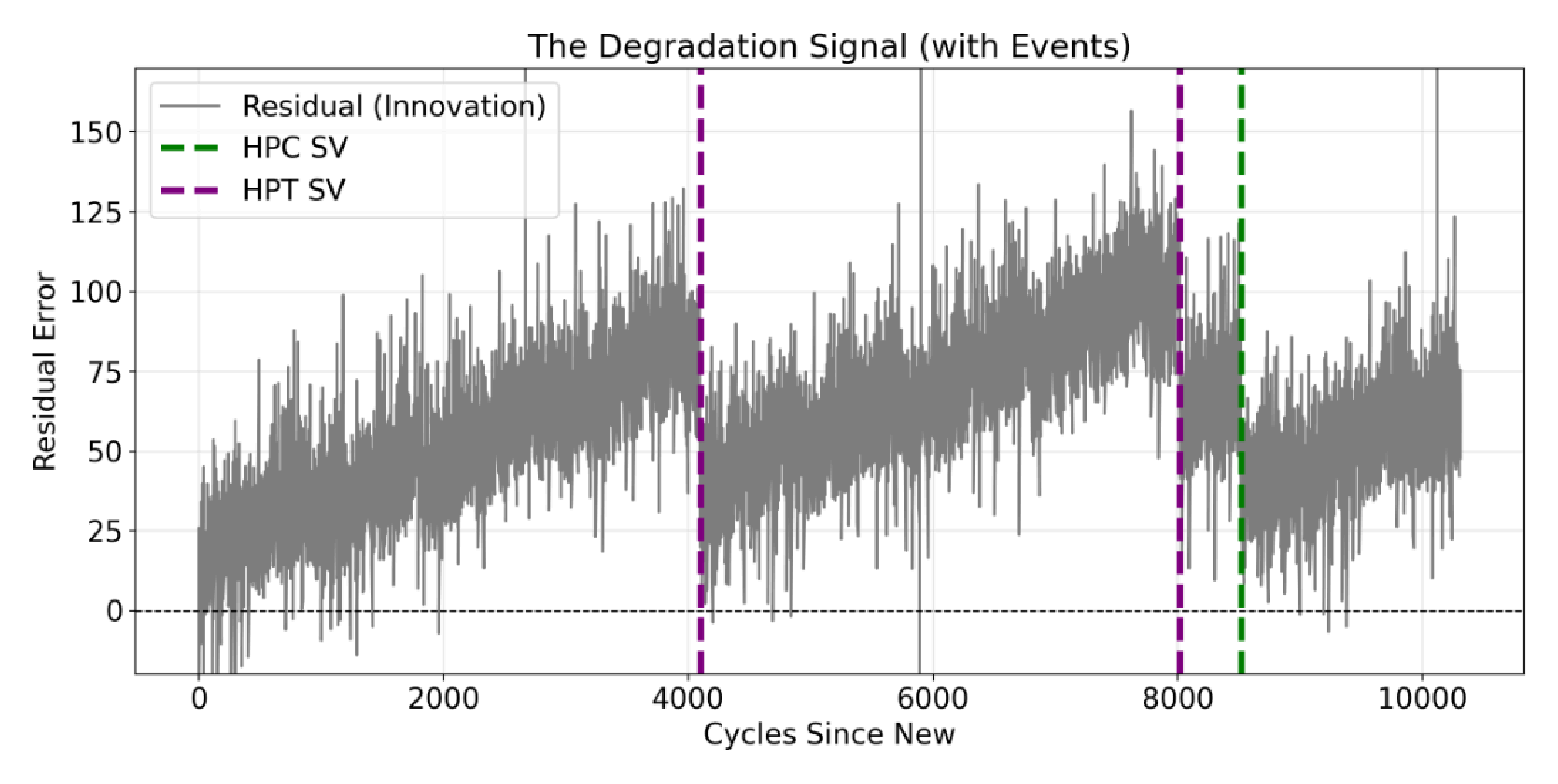

Figure 2: Filtered health state

Visualisation of health slope

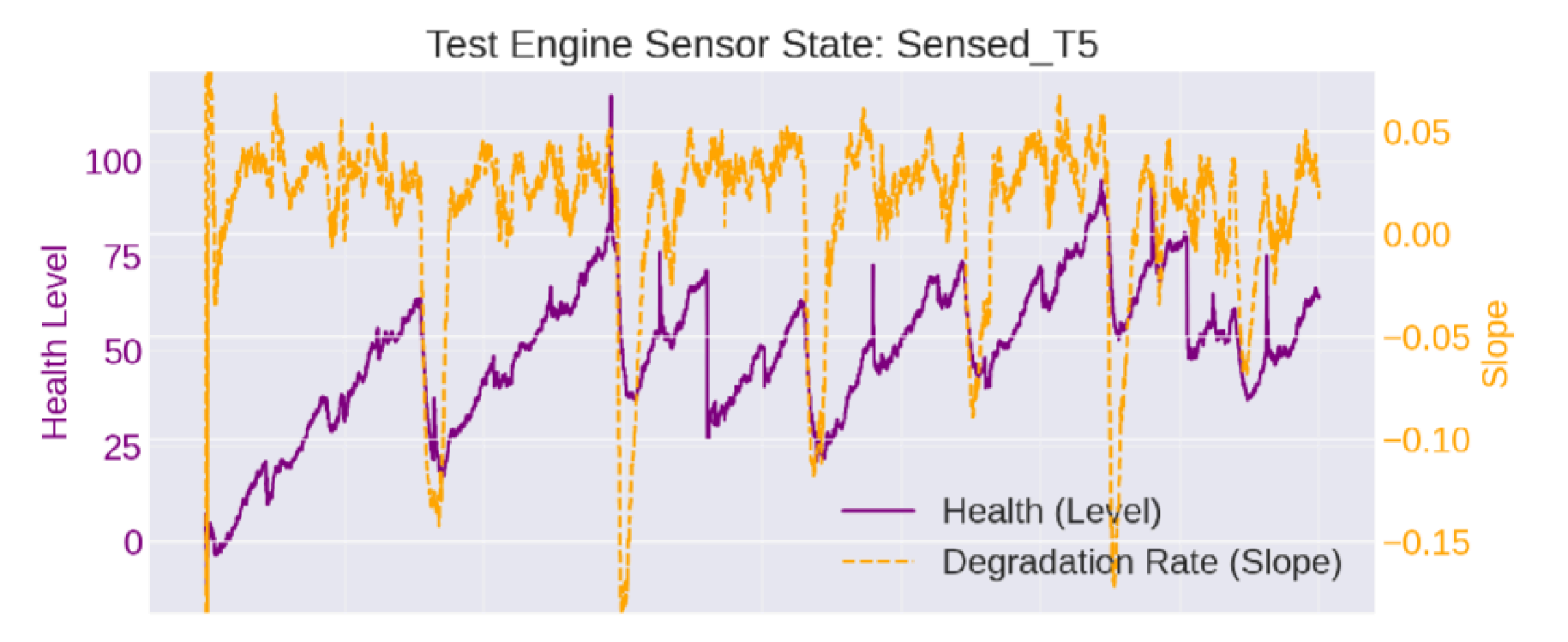

Figure 3: Filtered health slope

Predicting service events

How to get from these health states $\boldsymbol{h}_k$ and health slopes $\dot{\boldsymbol{h}}_k$ to predicted events?

Physical Approach:

- Define threshold h_{thresh} where we need repairs.
- Calculate time to failure $t_{\text{fail}} = \frac{h_{\text{thresh}} - h_k}{\dot{h}}$
- Issues:
  - h_{tresh} is not a simple constant. It changes repair to repair.
  - $\dot{h}$ is quite noisy.
  - We have N (for each sensor) health indices and slopes $\boldsymbol{h}, \dot{\boldsymbol{h}}$ are vectors. We have to figure out a way to fuse them.

Predicting service events

Machine Learning Approach:

- Use Kalman Filtered health states $\boldsymbol{h}_k$ and health slopes $\dot{\boldsymbol{h}}_k$ as features.
- Train a model to predict time to events t_{event} such that

$$f\left(\boldsymbol{h}_k, \dot{\boldsymbol{h}}_k\right) = t_{\text{event}}$$

- Only small amount of quality features so it is feasible to train a model.

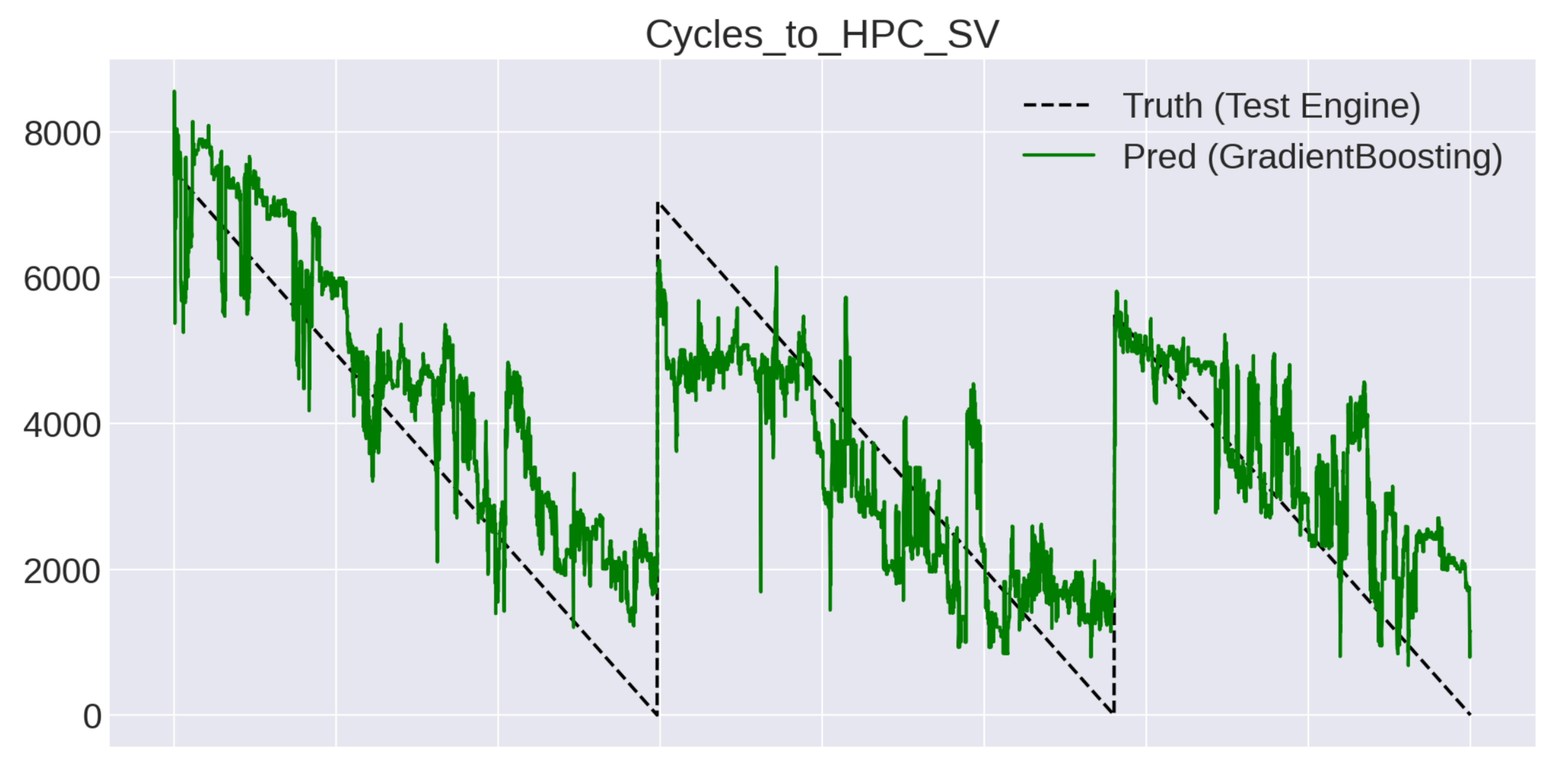

Figure 4: Results of Machine Learning Approach

Missed information

- We can still improve this model by using more information.
- Let's look at an example.
    - At t=100 we predict event in 100 steps.
    - At t=90 we predicted event in 105 steps.
    - We can combine these predictions to get better prediction.
    - This is implemented by a smoothing kalman filter over the predicted times.

Final smoothened prediction

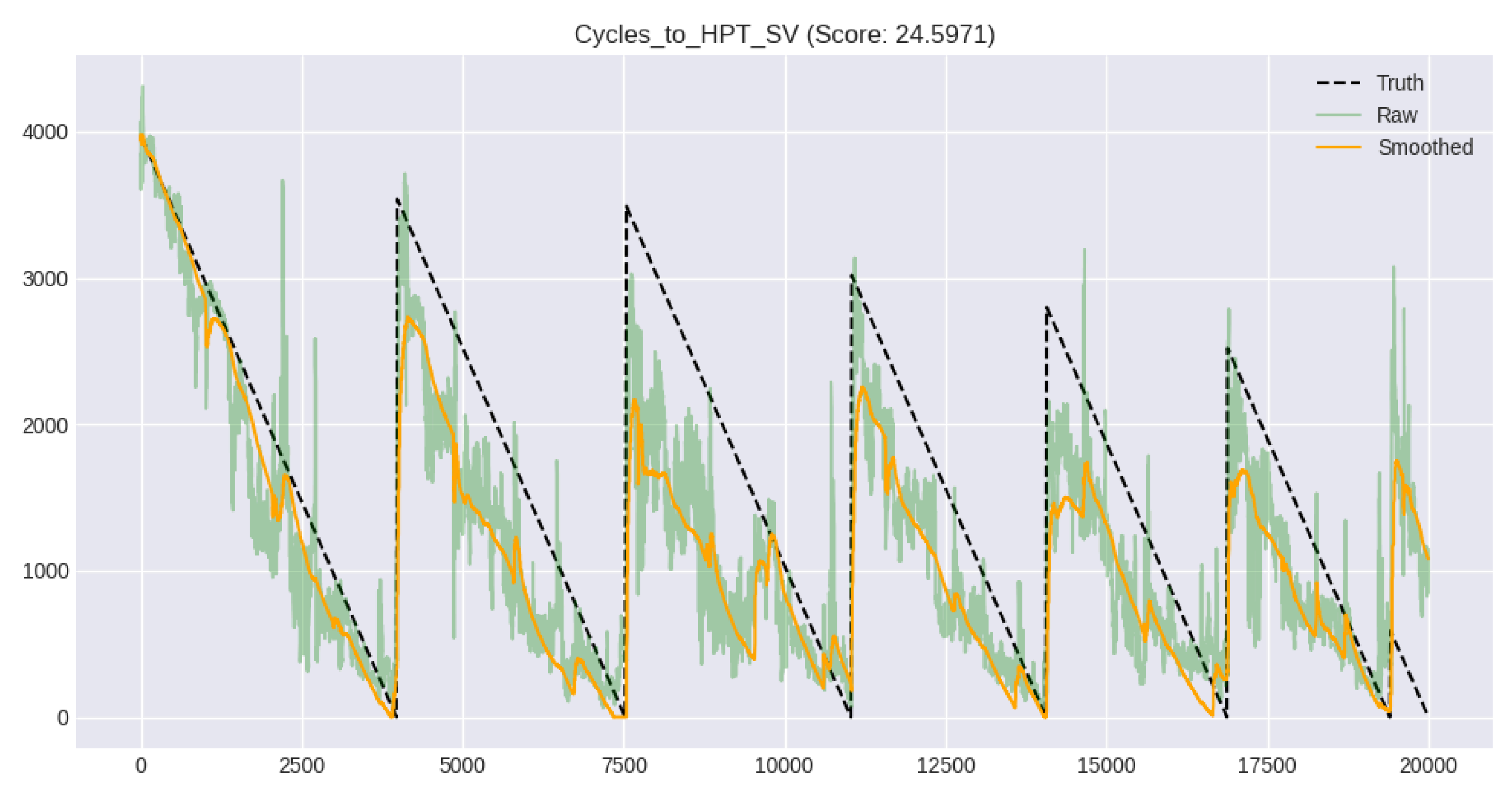

Figure 5: Final smoothened prediction

Final results

- Baseline tree model: 93.60
- Kalman filter model (without smoothing): 62.85
- Kalman filter model (with smoothing): 45.23
- Final model (with optimized hyperparams): 37.31

Standings at Close of Competition		
Team	**Validation Data Score**	**Testing Data Score**
SAM-IPA-1	47.54	37.11
lookhill	48.56	36.28
Justin_Boredom	49.3	37.22
CDTC	55.05	55.33
Q7	55.53	54.86
DeepFault	60.43	Not submitted
Whitebeard	60.95	62.85

Figure 6: Data challenge leaderboard