Journal of Open Aviation Science

JOAS

ARTICLE (***Pre-print***)

Fuel Consumption Estimation

Team Resourceful Quiver

Faculty of Aerospace Engineering, Delft University of Technology, Delft, Netherlands

Abstract

Accurately estimating aircraft fuel consumption from surveillance data is essential for assessing environmental impact and improving the efficiency of air traffic operations. However, many variables that determine fuel burn, such as aircraft mass, engine performance, and atmospheric conditions, are not directly observable from ADS-B. This study investigates the extent to which fuel consumption can be inferred statistically from ADS-B trajectories that are enriched with weather information, inferred mass, and physics-informed features. We construct a multi-stage preprocessing pipeline that includes ADS-B cleaning, interpolation, resampling, wind-data augmentation using ERA5, estimation of take-off weight, and derivation of fuel-flow and mass profiles based on the OpenAP performance model. Several feature groups are engineered, including statistical embeddings of ADS-B and ACARS variables, detailed vertical-rate phase decomposition, mass-related descriptors, and meteorological corrections. Using these features, we train a LightGBM regression model to predict fuel flow, which is then integrated to obtain segment-level fuel consumption. Experimental results on the challenge ranking dataset show that mass estimation, especially when using a heuristic interpolation between take-off and landing weight, substantially improves predictive accuracy. Models trained on fuel flow consistently perform better than those trained directly on fuel consumption, because the fuel-flow target exhibits lower skewness and better reflects the underlying physics. The final configuration achieves an RMSE below 200 on fuel-flow prediction, demonstrating the value of combining physics-informed features, statistical embeddings, and robust mass estimation. These findings highlight the importance of high-quality preprocessing and feature engineering for large-scale fuel-burn inference from sparse operational data.

Keywords: Fuel consumption estimation; ADS-B; ACARS; Aircraft mass estimation; OpenAP; LightGBM; Aviation environmental impact; Flight trajectory analytics; Fuel-flow; Feature engineering

Abbreviations: ADS-B: Automatic Dependent Surveillance–Broadcast, ACARS: Aircraft Communications Addressing and Reporting System, ATOW: Actual Take-Off Weight, TOW: Take-Off Weight, MTOW: Maximum Take-Off Weight, MLW: Maximum Landing Weight, OEW: Operational Empty Weight, TAS: True Airspeed, CAS: Calibrated Airspeed, QAR: Quick Access Recorder, ERA5: ECMWF Reanalysis Version 5, ECMWF: European Centre for Medium-Range Weather Forecasts, GBDT: Gradient Boosting Decision Trees, RMSE: Root Mean Squared Error

Disclaimer

This document is intended solely as documentation for the submissions of Team Resourceful Quiver to the PRC Data Challenge 2025.

1. Introduction

Aviation contributes significantly to global CO_2 emissions, which makes accurate fuel consumption estimation essential for assessing environmental impact and improving operational efficiency [1]. Fuel burn depends on many factors such as aircraft mass, engine characteristics, flight phase, and weather conditions, yet many of these variables are not directly observable from ADS-B data. This motivates the central question of this study: to what extent fuel consumption can be inferred statistically from ADS-B trajectories, supported by complementary information where available.

Previous research has explored both physics-based and data-driven approaches. Physics-based mod-

els, use aerodynamic principles and engine performance data. OpenAP builds on this foundation and incorporates statistical components in later versions. Data-driven systems such as Acropole combine QAR data, operational information, and machine learning methods to estimate fuel flow. Other studies, including neural-network based work by Jarry et. al [2], show that high-frequency flight data can reveal latent performance characteristics. Recent PRC Data Challenges also highlight the importance of mass estimation when predicting fuel burn.

In this challenge, we investigate a statistical approach to fuel consumption estimation using ADS-B trajectory data enriched with weather information, inferred mass, and physics-informed features. We evaluate how preprocessing choices, feature design, and regression models influence estimation accuracy.

This paper is organized as follows. Section 2 introduces the datasets. Section 3 describes the preprocessing pipeline. Section 4 presents feature engineering. Section 5 outlines the regression model. Section 6 discusses results. Section 7 concludes the study.

2. Data Description

2.1 Flight, Trajectory, and Fuel Data

Figure 1 shows samples of 1000 flights from the training set. As seen in the image, the flights spans almost worldwide with most data coming from Europe and North America. Also, we can see that the message update interval is not consistent, some has more update than the other.

Figure 1. Samples of 1000 flights from the training set.

Figure 2 shows an example of different variables in the training data plotted against time and the related fuel segment, with airport marked in green arrows. From this plot, we can already get some important insights about the dataset. First of all, some fuel segments in the same flight ID are identified before the the take-off time. This also happens in other flight ID and we also found that some fuel segments are after landing time. This also results in those segments outside the take-off to landing time to not having any data points at all and we only know the segment duration.

The next observation is the time window for the fuel vary in size which means the fuel consumption is widely distributed. When the time window is large, the fuel segment can be within two different flight phases, for instance climb and cruise, as seen from the altitude profile. This holds an important information since different flight phase has different fuel rate consumption according to OpenAP.

Figure 3 shows the histogram of the fuel consumption. As expected, due to the different time windows, the distribution is very wide, with the minimum recorded value is less than 0.5 kg, an average

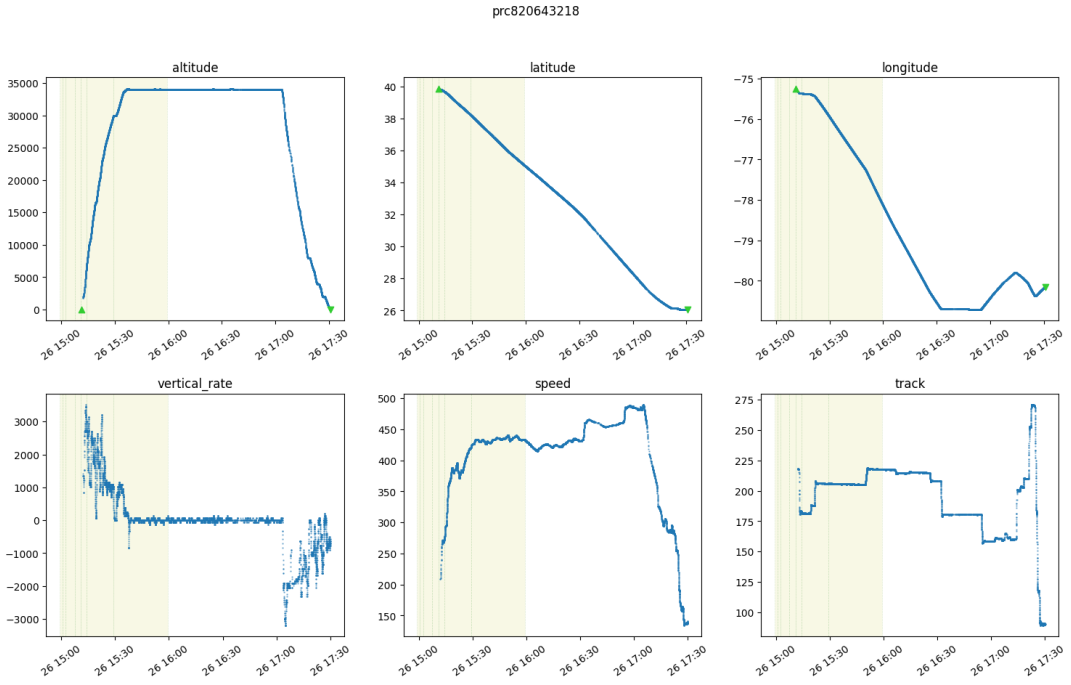

Figure 2. An example of time-series profiles of flight parameters including altitude, latitude, longitude, vertical rate, speed, and track. Green arrows mark airport arrival and departure points. Yellow shaded regions highlight the fuel segment.

of around 560 kg, and the maximum recorded value is more than 30000 kg.

From the literature (OpenAP, Acropole), what they predict is the fuel flow in kg/s. Since the calculation is straightforward, by dividing the fuel consumption in a specific segment to the segment duration, we plot it in Figure 4. The distribution is narrower for this feature because it is bounded by the physical attributes of the aircraft type. For instance, the minimum value in this case is less than 0.01 kg/s and the maximum value is around 16.5 kg/s. Both values are physically impossible for a passenger aircraft according to authors knowledge. By looking at the fuel flow, it is easier to separate potential noises in the data.

2.2 Take-off Weight Estimation Data

From previous literature, we understand that it is important to have the mass information of an aircraft to estimate the fuel consumption. This is logical because a heavier aircraft is more likely to burn more fuel than a lighter one since it requires more power to fly. For that reason, we also use take-off weight data for this challenge.

The take-off weight estimation data used in this study originates from the EUROCONTROL PRC 2024 Data Challenge [3], which provides an openly accessible dataset containing Actual Take-Off Weight (ATOW) information for commercial flights in Europe. The dataset was constructed from EUROCONTROL's Network Manager flight information and includes ATOW values derived from participating airlines, combined with ADS-B trajectories from the OpenSky Network and meteorological variables from ERA5. In total, the challenge offers 527,162 trajectories with corresponding take-off weight information, representing approximately 6.1% of all flights in European airspace during 2022. These data come with rich metadata, including flight identifiers, aircraft type, airport origin/destination, taxi times, and flown route distance, enabling the development of supervised

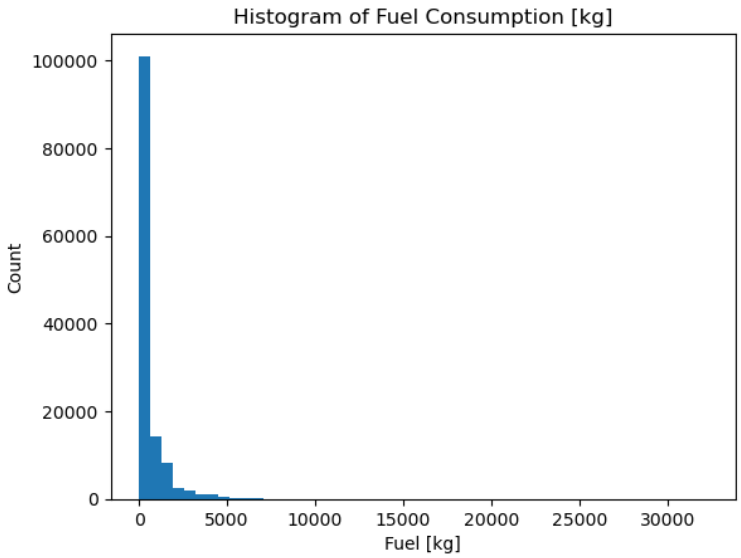

Figure 3. Fuel consumption in kg for the train set

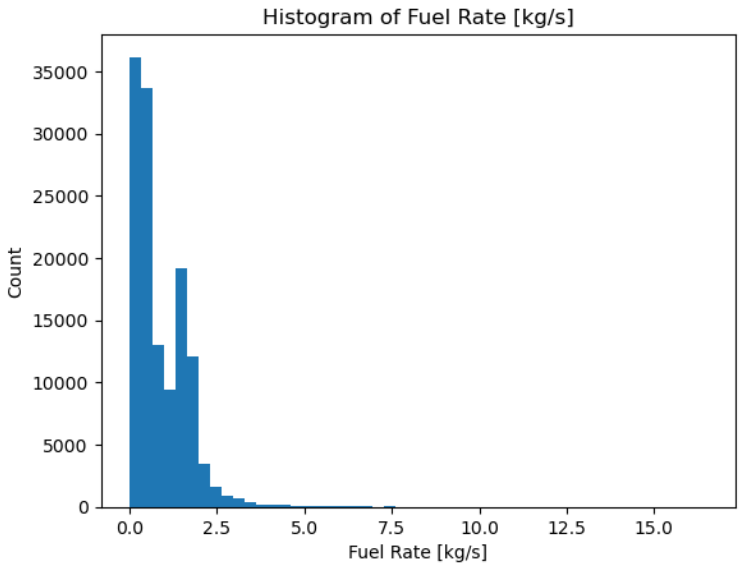

Figure 4. Fuel rate consumption in kg/s for the train set

learning models for take-off mass prediction.

2.3 Weather Data

We use the *fastmeteo* library to obtain the weather data per flight [4]. This library extracts ERA5 data from ECMWF [5] and interpolates it over the timestamps and spatial coordinates of the flight. For this challenge we extracted wind, air temperature, and humidity information, though the latter two were not used in the further model-building process. The u- and v-components of wind data were used to convert the groundspeed to true air speed (TAS) using track angle.

2.4 Complementary Data

We realized there's a missing altitude data in the airporrt data, so we add it there.

3. Preprocessing

We have built and tested several data preprocessing pipelines. In general, they follow the structure shown in Figure 5. These pipelines are designed to process trajectory data per flight. Here, in the **first** stage, we filter out the noise in ADS-B data, fill in the gaps using linear interpolation, and resample the data to equal time intervals. In the **second** stage, we add supplementary data that does not necessarily contribute to the final estimation model training, but is used to extract and compute some other important features. For example, such features include wind information, flight phases, airport information, time past since takeoff, etc. In the **third** stage, we fill in the missing TAS, Mach, and CAS data, which notably affect the accuracy of the final model. The **final** stage includes TOW estimation using most of the above-mentioned features and derivation of fuel flow and mass values per timestamp using OpenAP.

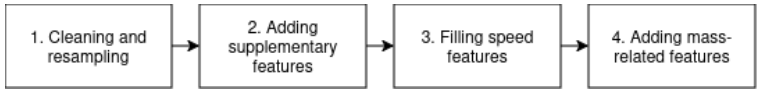

Figure 5. General data preprocessing pipeline.

3.1 Codebase Design

For preprocessing the data, we employ the *traffic* library [6], which enables data cleaning, resampling, and enhancing for each flight in multiple threads. The *Flight* class is extended with a child class *FlightProc* to facilitate modularized preprocessing steps. For the wind data augmentation, we used *fastmeteo* library to interpolate the filtered and resampled flights.

3.2 Preprocessing Procedure

The first preprocessing starts with loading the flight parquet files. We discovered that some flights have duplicated timestamps, latitude, and longitude. Those data points are firstly dropped. Secondly, some latitude values are outside the conventional range of -90 to 90 degrees. These data points are also removed after loading the parquet file. Subsequent steps generally follow the figure shown in 5, however, we also experimented with other steps that did not make it into the final preprocessing pipeline. Many of the flights do not have complete trajectories, and the fuel burn prediction window also covers those missing parts, we tried to augment the trajectories with the origin and destination airport locations, including the latitude, longitude, and elevation information while setting all other features to NaN. We also tried to complete the vertical profile of the trajectories which have missing climbing or descent segments using a predefined rate of climb and descent. However, while testing these features we saw significant higher RMSE on the rank set and they were subsequently removed

from the pipeline. We believe these steps added too much biased heuristics to the original data which degraded the prediction score.

Additionally, we also included mass estimation steps in the pipeline to facilitate the calculation of fuel flow rate for a given trajectory. A mass estimation model using LightGBM is trained on the dataset from last year's take-off weight estimation challenge. Another approach was estimating the mass of the aircraft based on the maximum take-off weight, maximum landing weight, and operational empty weight for a given aircraft type, as well as the percentage of time flown with respect to the airborne duration. The first method only predict the TOW and all subsequent mass are integrated using the fuel flow rate calculation from OpenAP. For some flights, the fuel flow rate numerical value explodes due to abnormal ADSB data, especially near the takeoff and landing time of a flight. Hence the mass calculation could have carried over error and eventually becomes negative. The second method mitigated the negative mass issue by providing less accurate TOW estimation, however, we see improved ranking score using this approach.

Lastly, meteorological data was used to enriched the TAS feature. After pre-syncing the ERA5 data using fastmeteo, u and v components of wind are interpolated for each data point and a computed TAS is analytically calculated based off the groundspeed.

4. Feature Engineering

4.1 Basic Features

We define the basic features as the set containing segment duration, segment distance, flight duration, full-flight distance, aircraft type, and phase. We refer to them as basic features because they serve as the core inputs for all regression models.

Segment duration is the time difference between the start and end of a fuel segment and represents how long the aircraft operates within that segment. Segment distance is the distance covered during the same period. It is computed by taking the minimum and maximum values of the distance variable within the segment timestamps and subtracting them. The distance variable itself is constructed as the cumulative sum of great-circle distances between consecutive trajectory points, which ensures consistency even when raw distance is not directly available.

Flight duration is obtained from the time difference between takeoff and landing, and full-flight distance is taken as the maximum value of the cumulative distance over the whole flight.

The last two features, aircraft type and phase, are categorical. Aircraft type is taken directly from the challenge dataset, while the phase is derived using a function from the Traffic library [6], which assigns flight phases based on the aircraft trajectory data.

4.2 ADS-B Data Features

ADS-B data features, or simply ADS-B features, are variables derived from ADS-B data. These include groundspeed, track, vertical rate, altitude, latitude, longitude, distance, cumulative distance, and timestamp. If ADS-B data does not provide these variables, we obtain them from ACARS data. For simplicity, we still categorize them as ADS-B features.

Within a flight, these variables change over time. To use them in a tabular model, we convert the temporal signals into fixed-size features. We do this by computing summary statistics such as mean, standard deviation, minimum, and maximum for each variable. We refer to this process as statistical embedding, and we apply it not only to ADS-B features but also to other temporal features.

Statistical embedding is important because it provides context for each variable. Using only the mean can hide important differences in flight behavior. For example, a flight in stable level flight

and a flight that climbs and descends frequently may have the same mean vertical rate, but their standard deviation, minimum, and maximum values will differ. These differences capture the actual dynamics of the flight.

For model training, we use only the statistical embeddings of groundspeed, track, vertical rate, and altitude. Although we can compute statistical embeddings of distance, cumulative distance, timestamp, latitude, and longitude, we exclude them because they do not contribute directly to fuel consumption from a physical perspective.

Based on experimentation, we use the mean and standard deviation for groundspeed and track. For vertical rate, we use mean, standard deviation, minimum, and maximum. For altitude, we use only the mean.

4.3 ACARS Data Features

ACARS data features, or ACARS features, include true airspeed (TAS), calibrated airspeed (CAS), and Mach number. Since these variables also appear in the trajectory data, we compute statistical embeddings for them as well.

The data are often sparse, and these three variables are not always available at every timestamp. To address this, we infer the missing values from any ACARS readings that are present. For instance, if TAS is missing but CAS is available, we convert CAS to TAS using the `cas2tas` function from the OpenAP library. If CAS is not available but Mach is, we convert Mach to TAS using `mach2tas`. If both are missing, we fall back to groundspeed as an approximation. After estimating TAS, we recompute Mach and CAS from the inferred TAS so that all three variables remain consistent.

4.4 Wind Data Features

The wind-related features are derived from meteorological data. We compute them using the data described in Section 2.3 together with the groundspeed data. As described above, we calculate TAS using the u and v components of the wind, and we convert groundspeed to TAS using the track angle. We then calculate the corresponding CAS and Mach number. We refer to these variables as wind features.

4.5 Mass-related Features

The mass-related features are defined as the set of variables that use takeoff weight information as input. First, we estimate the takeoff weight of the aircraft using PRC 2024 data [3]. We train a simplified estimator based on LightGBM, which achieves an estimation RMSE of approximately 3000 kg. It is important to note that the training data consist of European flights in 2024, and the estimation model does not cover all aircraft types in the current paper.

Table 1 provides an overview of the aircraft types available in OpenAP and in the PRC Challenge 2024. Missing type codes may cause higher uncertainty and error in the performance of our TOW and fuel-flow estimation models.

Table 1. Aircraft types available

This challenge	A20N	A21N	A306	A318	A319	A320	A321	A332	A333	A359	A388	B38M	B39M	B734	B737	B738	B739	B744	B748	B752	B763	B772	B77L	B77W	B788	B789	MD11
OpenAP	(✓)	✓	(✓)	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	(✓)	✓	✓	✓	(✓)
PRC 2024	✓	✓			✓	✓	✓	✓	✓	✓		✓	✓		✓	✓	✓			✓	✓	✓		✓	✓	✓	

(✓) using a similar aircraft's properties in OpenAP

Since OpenAP's library require mass as an input for the fuel flow estimation, we need to estimate the mass at each point in time. We provide two ways of calculating the mass at each point, the first one is called recursive mass and the second one is heurisitc mass. They will be explained shortly. These mass-related variables are converted to tabular data by taking the mean and standard deviation within the prediction window, the same statistical embedding.

In addition, we include MTOW and OEW features sourced from the OpenAP library and other available sources (for example, the Eurocontrol Aircraft Performance Database and SKYbrary). Thus, the mass-related feature set contains MTOW, OEW, estimated TOW, the mean and standard deviation of fuel-flow estimates, and the mean and standard deviation of mass estimates.

4.5.1 Recursive Mass

Using the OpenAP library [7], we calculate the mass and fuel flow at each data point, given the initial mass. This process is partially recursive because fuel flow depends on the current mass, which is calculated using the OpenAP *Fuel* module, and the mass is updated as the initial mass minus the cumulative fuel flow:

$$m_n = m_0 - \sum_{i}^{n} ff_i \cdot \Delta t_i, \qquad \forall n \in [0, 1, \ldots, k] \tag{1}$$

where m_n is the mass in kg at timestamp n and m_0 corresponds to TOW; ff is the fuel flow in kg/s; and Δt is the time difference between timestamps in s. We find that five iterations are sufficient for convergence to stable mass values.

4.5.2 Heuristic Mass

For the heuristic mass estimation, we follow a simple deterministic interpolation between an assumed take-off weight (TOW) and an assumed landing weight (LW). Unlike the recursive method, no fuel-flow model is used; instead, the aircraft mass is approximated as a linear function of the fraction of the flight completed in time.

Using aircraft properties from the OpenAP [7] *Prop* module, we first define representative weights based on type-specific limits. The take-off weight is set to a constant proportion of the maximum take-off weight (MTOW), while the landing weight is taken as the midpoint between the maximum landing weight (MLW) and the operational empty weight (OEW):

$$m_{\mathrm{TO}} = 0.8 \cdot \mathrm{MTOW}, \qquad m_{\mathrm{LD}} = 0.5 \cdot (\mathrm{MLW} + \mathrm{OEW}). \tag{2}$$

Let t_{TO} and t_{LD} denote the take-off and landing timestamps respectively. For each data point with timestamp t_n, we compute the proportion of flight time completed as

$$p_n = \frac{t_n - t_{\mathrm{TO}}}{t_{\mathrm{LD}} - t_{\mathrm{TO}}}, \tag{3}$$

where p_n is clipped to the interval $[0, 1]$ to avoid extrapolation for points prior to take-off or after landing. The mass estimate then becomes a linear interpolation between m_{TO} and m_{LD}:

$$m_n = m_{\mathrm{TO}} - p_n \cdot \left(m_{\mathrm{TO}} - m_{\mathrm{LD}}\right), \qquad \forall n \in [0, 1, \ldots, k]. \tag{4}$$

This method provides a fast, model-free approximation that captures the expected monotonic decrease in mass across the flight, though without accounting for variations in fuel burn due to speed, altitude, or aircraft configuration. Our estimated take off weight is also not yet included here, this is future work.

4.6 Detailed Vertical Rate Features

From Figure 2, we observe that a fuel segment can span a wide time window. This means that it can cover multiple scenarios such as only climb, climb followed by cruise, only cruise, cruise followed by descent, or only descent. The fuel rates of these different flight phases are distinct. To capture the effect of different flight phases within a segment, we construct a detailed vertical rate feature.

This detailed vertical rate feature is created by splitting each fuel segment into ten parts of equal time length. For each part, we compute the mean and standard deviation of the vertical rate. This procedure yields 20 additional features per segment.

5. Regression Model

Many approaches exist for predicting fuel burn from trajectory data. Acropole, for example, applies a neural-network regression model trained on Quick Access Recorder (QAR) data to infer fuel flow from ADS-B inputs [8]. We also evaluated several Gradient Boosting Decision Tree (GBDT) methods, including Random Forest, CatBoost, and XGBoost. LightGBM was selected because it is computationally efficient, uses histogram-based splits, and handles missing values natively. The baseline parameters used in our experiments are listed in Table 2.

Table 2. Baseline LightGBM parameters

Parameter	Value
n_estimators	5000
learning_rate	0.01
subsample	0.8
colsample_bytree	0.8
reg_lambda	0.5
reg_alpha	0.1
metric	rmse
random_state	46

Since LightGBM uses histogram-based binning of the features, we train the model on the fuel flow rather than directly on the fuel consumption. Although the binning process does not depend on the scale of the target variable, the distribution of fuel flow is considerably tighter and less skewed than the distribution of fuel consumption, which results in a more stable learning target.

In addition, we remove fuel-flow values below 0.05 kg/s and above 6.5 kg/s. These extremes are physically implausible for the aircraft types in the dataset and typically originate from noisy or corrupted trajectory segments. Filtering them ensures that the model is trained within a realistic operating range and prevents overfitting to artefacts outside the feasible bounds.

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

We use RMSE as the loss function because it penalizes large deviations more strongly than MAE, which is important for capturing high fuel-flow regimes accurately. RMSE is also the official evaluation metric of the challenge, ensuring consistency between training and scoring.

6. Results and Discussion

We define a benchmark model that uses ADS-B, ACARS, and the detailed vertical-rate split features, trained on the filtered fuel-flow target. All configurations discussed below include the same base features and the preprocessing pipeline described earlier. The tables in this section present only the differences in fuel-flow filtering and feature combinations.

We also did a k-fold cross validation, using data that is not randomly splitted to avoid leakage from two identical flight id. For the simplicity of this report, and since the RMSE from the ranking set already provide a more valuable insight, we omitted it from here.

Table 3. Fuel flow filter benchmarking on ADSB + ACARS + Split

Lower Limit	Upper Limit	RMSE
-1	1000	228.05
-1	6.5	205.00
0.05	6.5	205.85

6.1 Fuel-flow filtering

We apply a heuristic fuel-flow filter with bounds of 0.05 to 6.5 kg/s, which yields an RMSE of 205.85 on the ranking set. To evaluate the sensitivity of the model to this filtering, we compare the benchmark with two alternative settings: (i) removing both limits using very wide bounds, and (ii) keeping only the upper limit while removing the lower limit.

Table 3 shows that removing both limits leads to a substantial degradation in RMSE. Using only the upper limit results in a slightly lower RMSE than the benchmark. This indicates that unrealistic low fuel-flow values introduce more noise than unrealistic high values, and that filtering is essential for stable model performance.

6.2 Feature-set comparison

Table 4 summarises the RMSE for different feature combinations. Starting from the benchmark (ADSB + ACARS + Vrate split), adding wind data provides a small improvement. However, replacing ACARS with wind causes a notable deterioration. This performance is similar to models using only ADS-B or ADS-B with vertical-rate splitting. In contrast, using ADS-B and ACARS together already improves the RMSE, and combining them with the Vrate split yields a further reduction.

Next, we extend the benchmark with mass-related features using the recursive mass approach. This full configuration includes MTOW, OEW, estimated takeoff weight, estimated fuel flow, and estimated mass statistics. The resulting model achieves an RMSE of 203.75, an improvement over the benchmark. Removing any of the mass-related components worsens performance, which shows that these features work best when used together.

We also examine redundancy between ACARS and wind data. When mass features are included, the model that retains ACARS but removes wind performs better than using both sources. This is different from the no-mass case, where ADS-B, ACARS, wind, and Vrate split together produce the best result, although the improvement is small and may fall within noise.

We further test the heuristic mass method. Since earlier results show that mass-related features act synergistically, we keep all mass variables instead of removing them individually. The full configuration reaches an RMSE of 201.04, which is lower than all recursive-mass models. Removing ACARS

Table 4. Model configuration and corresponding RMSE

Description	ADS-B	ACARS	Wind	Vrate Split	MTOW + OEW	TOW Est	FF Est	Mass Est	RMSE
Fuel flow [kg/s], no Mass									
Benchmark	✓	✓		✓					205.85
Benchmark + Wind	✓	✓	✓	✓					205.25
ADSB + Wind + Split	✓		✓	✓					212.80
ADSB only	✓								212.67
ADSB + Split	✓			✓					211.42
ADSB + ACARS	✓	✓							210.71
Fuel flow [kg/s], with Recursive Mass									
Full features	✓	✓	✓	✓	✓	✓	✓	✓	203.75
No Mass Est	✓	✓	✓	✓	✓	✓	✓		206.22
No FF + Mass Est	✓	✓	✓	✓	✓	✓			206.06
No TOW/FF/Mass Est	✓	✓	✓	✓	✓				205.95
No ACARS	✓		✓	✓	✓	✓	✓	✓	206.48
No Wind	✓	✓		✓	✓	✓	✓	✓	203.09
Fuel flow [kg/s], with Heuristic Mass									
Full features	✓	✓	✓	✓	✓	✓	✓	✓	201.04
No ACARS	✓		✓	✓	✓	✓	✓	✓	208.71
No Wind	✓	✓		✓	✓	✓	✓	✓	199.91
Fuel estimate [kg], with Recursive Mass									
Full features	✓	✓	✓	✓	✓	✓	✓	✓	220.56
ADSB + ACARS	✓	✓							219.13
ADSB + ACARS + Split	✓	✓		✓					218.47

in this setting significantly worsens performance (RMSE 208.71), while removing wind improves it (RMSE 199.91). This again suggests that ACARS provides more useful information than wind data.

6.3 Training on fuel consumption

Finally, we train models directly on fuel consumption rather than fuel flow, while still applying the 0.05–6.5 kg/s filter before converting flow to segment-level consumption. We test a few representative feature sets. The full model produces an RMSE of 220.56, while a reduced ADS-B + ACARS model reaches 219.13. Adding the Vrate split further lowers the RMSE to 218.47. These results confirm that training on fuel flow yields substantially better performance than training directly on fuel consumption, even when using the same input features.

6.4 Is mass estimation important?

The benchmark results show that models without mass-related features are limited in accuracy. Using only ADS-B, ACARS, and vertical-rate split features, the best RMSE achieved is 205.25, and removing wind information yields 205.85. Once mass-related features are added, the performance improves. The recursive mass approach reduces the RMSE to 203.09, while the heuristic mass method results in a much larger improvement, with RMSE values reaching 199.91.

These results indicate that mass estimation provides information that the model cannot fully infer from kinematic or atmospheric variables. Aircraft mass influences the required thrust and therefore the fuel burn. Without mass features, the regression model must approximate this latent variable indirectly, which limits its predictive capability.

The two mass-estimation strategies illustrate this effect clearly. The recursive mass method uses OpenAP fuel-flow calculations combined with an estimated take-off weight. Since mass at each timestamp is obtained by subtracting cumulative fuel burn, any early inaccuracies in estimated fuel flow influence all later values. As a result, errors propagate throughout the flight and can become significant. Additionally, some flight with less to no data during take-off might disturb the estimation significantly. Missing the trajectory information early in the flight results in error accumulation further ahead the trajectory since it is recursively calculated.

In contrast, the heuristic mass method avoids these issues by assigning mass through linear interpolation between representative take-off and landing weights. Although this approach does not explicitly model the physics of fuel burn, it provides smooth and stable mass estimates that are not influenced by noisy fuel-flow calculations. This estimate also provides a bounded guarantee that is physically realistic, even when there's no trajectory data at all. The resulting features are easier for the regression model to use, and this potentially explains the larger improvement in RMSE.

Overall, even approximate mass information helps the model capture a key driver of fuel consumption, and the results demonstrate that including such information is beneficial for fuel-flow prediction.

In the future, further work on mass estimation is required to better assess the importance of this feature. A first improvement is to replace the current assumption of using 80% of the maximum take-off weight (MTOW) with the estimated take-off weight obtained from the TOW model. A next step is to develop a more accurate take-off weight estimator and evaluate how improvements in TOW prediction affect the overall fuel-burn estimation performance.

7. Conclusion

In this challenge, we discovered that data preprocessing has a big impact on the prediction quality. Attempts to complete the trajectory always lead to worse result, though this maybe due to the biased heuristic we provided that does not follow operational and realistic scenarios. Keeping as much original raw data as possible while filtering out obvious outliers generally lead to better result. However, some outliers, such as GPS spoofed or jammed sections, are difficult to classify and out of the scope of our capabilities. We found that leaving them untouched yields better result.

ACARS data, albeit sparse, provide valuable insights in the prediction process. Meteorological data enrichment does not lead to significant improvement in prediction accuracy. One of the reasons we believe is that the scale of meteorological data and trajectory data operate on two different scales. Any interpolation on the sparse wind grid lose too much local information on where the aircraft is actually flying, information that is instead directly measured and broadcast through ACARS. Additionally, the massive range of meteorological data makes iterating on the preprocessing pipeline slow.

The final feature set was obtained in an iterative process that combined physical reasoning with empirical validation on the k-fold cross validation (not shown in this documentation) and ranking set. Starting from all engineered feature groups, we successively removed or simplified features that did not provide a clear and robust improvement in RMSE, or that were judged to be redundant with more informative variables. At each step, we trained a LightGBM model with identical parameters

and compared its performance to the current baseline, which allowed us to isolate the contribution of each feature group in a controlled manner.

In this process, the statistical embeddings of ADS-B and ACARS variables formed the backbone of the model, since they directly describe the aircraft kinematics, airspeed, and their variations. Detailed vertical-rate features were retained because they consistently captured differences in the mix of climb, cruise, and descent within a segment and led to stable improvements in RMSE. Meteorological features, on the other hand, were kept only in a limited form. While wind-corrected TAS can be useful, most wind variables provided little additional benefit compared to ACARS, and in some configurations even degraded performance, which we attribute to interpolation noise and scale mismatch between gridded weather data and the trajectory.

Mass-related variables were included only after a similar ablation study. Models that used recursive or heuristic mass estimates systematically outperformed otherwise identical models without mass, and removing any of the mass components (estimated TOW, MTOW and OEW, mass and fuel-flow statistics) resulted in a loss of accuracy. As a result, the final model uses the full mass feature set together with ADS-B, ACARS, and detailed vertical-rate embeddings, while higher-dimensional or weakly contributing features are omitted.

Reproducibility statement

For the code, please refer to https://github.com/meldorr/PRC-2025/tree/main

Information on how to reproduce this research, including access to: 1) source code related to the research, 2) source code for the figures, 3) source code/data for the tables when applicable.

References

[1] Roger Teoh, Zebediah Engberg, Ulrich Schumann, Christiane Voigt, Marc Shapiro, Susanne Rohs, and Marc E. J. Stettler. "Global aviation contrail climate effects from 2019 to 2021". en. In: *Atmospheric Chemistry and Physics* 24.10 (May 2024), pp. 6071–6093. ISSN: 1680-7324. DOI: 10.5194/acp-24-6071-2024. URL: https://acp.copernicus.org/articles/24/6071/2024/ (visited on 11/29/2025).

[2] Gabriel Jarry, Ramon Dalmau, Philippe Very, and Junzi Sun. "On the generalization properties of deep learning for aircraft fuel flow estimation models". In: *Transportation Research Part C: Emerging Technologies* 176 (2025), p. 105143. ISSN: 0968-090X. DOI: https://doi.org/10.1016/j.trc.2025.105143. URL: https://www.sciencedirect.com/science/article/pii/S0968090X25001470.

[3] Enrico Spinielli, Junzi Sun, Martin Strohmeier, Xavier Olive, Quinten Goens, Rainer Koelle, Allan Tart, and John Fitzgerald. "Aircraft Takeoff Weight Estimation: The EUROCONTROL PRC 2024 Data Challenge". In: *Journal of Open Aviation Science* 3.2 (July 2025). DOI: 10.59490/joas.2025.8252. URL: https://journals.open.tudelft.nl/joas/article/view/8252.

[4] Junzi Sun and Esther J. Roosenbrand. "Fast contrail estimation with OpenSky data". In: *ournal of Open Aviation Science* (2024). DOI: https://doi.org/10.59490/joas.2023.7264.

[5] Hans Hersbach, Bill Bell, Paul Berrisford, Gionata Biavati, András Horányi, Joaquín Muñoz Sabater, Julien Nicolas, Carole Peubey, Raluca Radu, Iryna Rozum, et al. "ERA5 hourly data on pressure levels from 1940 to present". In: *Copernicus climate change service (c3s) climate data store (cds)* (2023). DOI: 10.24381/cds.bd0915c6.

[6] Xavier Olive. "traffic, a toolbox for processing and analysing air traffic data". In: *Journal of Open Source Software* 4 (2019), p. 1518. ISSN: 2475-9066. DOI: 10.21105/joss.01518.

[7] Junzi Sun, Jacco M. Hoekstra, and Joost Ellerbroek. "OpenAP: An Open-Source Aircraft Performance Model for Air Transportation Studies and Simulations". In: *Aerospace* 7.8 (2020). ISSN: 2226-4310. DOI: 10.3390/aerospace7080104. URL: https://www.mdpi.com/2226-4310/7/8/104.

[8] Gabriel Jarry, Daniel Delahaye, and Christophe Hurter. “Towards aircraft generic Quick Access Recorder fuel flow regression models for ADS-B data”. In: *International Conference on Research in Air Transportation*. 2024.