

ARTICLE

Fuel Burn Prediction from Open Flight Trajectory and Meteorological Data for the PRC Data Challenge 2025

Rade Kačar ^{*},¹ and Darko Čulibrk ^{*},²

¹University of Belgrade, Faculty of Transport and Traffic Engineering, Belgrade, Serbia

²Planning and Development of ISP Network Mtel a.d. Banja Luka, Banja Luka, Bosnia and Herzegovina

*Corresponding author: rade.kacar@sf.bg.ac.rs; darko.culibrk@mtel.ba

(Received: 31 Mar 2026; Revised: 12 Jul 2026; Accepted: 3 Aug 2026; Published: 26 Aug 2026)

(Editor: Enrico Spinielli; Reviewers: Quinten Goens and Allan Tart)

Abstract

This paper describes a segment-level fuel burn prediction system developed for the PRC Data Challenge 2025, operating on open aviation data sources: ADS-B and ACARS trajectories, ERA5 pressure-level re-analysis, METAR surface observations, and FAA aircraft performance records. The dataset comprises 131,530 labeled segments derived from 11,037 European flights (April–September 2025). Rather than applying machine learning directly to raw trajectory statistics, the methodology constructs a 73-dimensional feature space that encodes physically motivated intermediate quantities: ERA5 wind-corrected true airspeed, OpenAP-derived drag and thrust coefficients, total energy model mass estimates for climb and descent phases, and phase-specific fuel-flow surrogates identified via the traffic library phase detector. Gradient boosting regressors (XGBoost, LightGBM) with Optuna-tuned hyperparameters are trained on this feature space using K-fold cross-validation. XGBoost achieves a cross-validation RMSE of 232.76 kg; LightGBM reaches 243.91 kg. The best submission to the challenge blind test set scores 227.32 kg RMSE. The primary methodological distinction from prior purely statistical approaches is the systematic substitution of latent physical variables — mass, airspeed, phase-resolved fuel flow — in place of raw kinematic observables, without requiring access to proprietary flight data recorder or BADA performance data.

Keywords: fuel burn prediction; aviation analytics; trajectory mining; gradient boosting; physics-informed machine learning

1. INTRODUCTION

Fuel burn prediction from surveillance data is an operationally relevant problem in aviation: it informs airline cost modeling, trajectory efficiency benchmarking, and emissions inventory. Approximate fuel usage can be estimated from aircraft type, great-circle distance, and standard operational assumptions, but actual fuel burn depends on a substantially richer set of factors, including trajectory geometry, atmospheric state, operational phase transitions, and the fidelity of available surveillance observations. This dependence makes fuel burn a natural regression target for data-driven modeling, provided that the feature representation reflects the underlying physical and operational structure of flight. Physics-based methods, such as those using BADA (Base of Aircraft Data) performance tables [1, 2] or manufacturer-specific flight performance models [3], can achieve fuel estimation errors below 5% when applied to high-quality trajectory data with known takeoff mass, but they require either proprietary performance data or iterative mass initialization that is sensitive to initial conditions. Data-driven methods applied to QAR (Quick Access Recorder) data [4, 5, 6] exploit high-

resolution onboard measurements that are not publicly accessible. Prior work combining ADS-B surveillance with ERA5 meteorological data and open performance models has demonstrated that full-trajectory fuel estimation errors below 2% are achievable for a small number of validated flights [7], but large-scale application across diverse aircraft types and route structures, using segment-level target labels rather than total-trip fuel, remains less studied. No prior work in the open literature has systematically evaluated physics-informed feature engineering — specifically, total energy model mass estimation and phase-resolved fuel-flow surrogates derived from OpenAP — within a gradient boosting framework operating on heterogeneous ADS-B/ACARS segments at the scale of tens of thousands of flights. The PRC Data Challenge 2025 defines a segment-level prediction task over 11,037 European flights, providing ADS-B (Automatic Dependent Surveillance-Broadcast) and ACARS (Aircraft Communications, Addressing and Reporting System) trajectories. This work addresses that task through a five-stage pipeline: data ingestion and validation, trajectory-based segment extraction, meteorological augmentation, physics-informed feature derivation, and gradient boosting regression with automated hyperparameter optimization. The feature derivation stage, which estimates true airspeed from ERA5 wind-corrected groundspeed, identifies climb, cruise, and descent phases, and solves a total energy balance for per-phase aircraft mass, constitutes the primary methodological contribution of this work relative to approaches that treat segment-level statistical summaries as direct model inputs. The remainder of this paper describes the data sources, feature engineering procedures, weather enrichment process, phase-aware performance modeling, and machine learning configuration, followed by empirical results and directions for further development.

2. RELATED WORK

Fuel burn estimation from surveillance data has been investigated through two complementary paradigms: physics-based methods that rely on aircraft performance models [1, 2, 8] and data-driven methods that apply statistical or machine learning techniques to operational flight records [9, 10].

The physics-based approach derives fuel flow from the equations of motion under a point-mass assumption. At the institutional level, the FAA (Federal Aviation Administration) Aviation Environmental Design Tool (AEDT) consolidates fuel consumption, emissions, and noise modeling into a single framework, but its core algorithm relies on thrust profiles from the flight data recorder or on standard thrust schedules, limiting its applicability when FDR (Flight Data Recorder) data are unavailable [8]. Chatterji [11] formalized this procedure by estimating aircraft and wind states, lift, drag, and thrust from 4D flight track data, using BADA [1] fuel-flow and drag models. The method achieved fuel burn estimates within 1% of flight test data for a single validation flight with known takeoff weight, and within 5.4% when using lower-quality radar-based position data with iterative weight estimation. Dalmau et al. [3] extended this methodology to a larger-scale validation of 2448 descent operations, comparing fuel estimates from both radar tracks and FDR trajectories against recorded FDR fuel consumption. Using an aircraft manufacturer performance model (PEP), the mean estimation error was 4.8% from FDR tracks and 7.8% from surveillance radar, whereas BADA 3.6 [2] underestimated fuel consumption by 18.7%, indicating the sensitivity of physics-based methods to the fidelity of the underlying performance model.

Krajčec Nikolić et al. [7] performed a systematic assessment of fuel burn estimation using ADS-B data, ERA5 reanalysis meteorological fields, and BADA 3 aircraft performance data for the entire airborne trajectory including non-clean configurations. Validation against FDR for five commercial flights of an Airbus A320-family aircraft yielded an average fuel estimation error of 1.2% with a standard deviation of 1.3%. The study demonstrated that the combination of ADS-B and ERA5 data provides sufficient input quality for phase-resolved fuel burn calculation when supplemented with configuration scheduling.

A separate line of research addresses aircraft performance modeling using open data. Sun et al. [12] developed OpenAP, an open-source aircraft performance model constructed from ADS-B surveillance data and published literature models. The theoretical foundations of OpenAP, including the data-driven methods for kinematic parameter extraction from ADS-B surveillance, drag polar estimation, and fuel-flow modeling, are documented in Sun's doctoral dissertation [13], which also presents validation of the open model against BADA and operational flight data. OpenAP provides drag polar, thrust, and fuel-flow submodels that do not require proprietary BADA licensing, thereby enabling redistribution and reproducibility. Filippone et al. [14] integrated ADS-B and Mode-S data with a flight performance simulation framework to estimate aviation emissions for over 2500 trajectories of multiple aircraft types, demonstrating that real-time surveillance data can replace simulated trajectories in emission inventory calculations. This work was further extended to cover a broader range of aircraft types, including turboprops and business jets [15].

Machine learning methods for fuel consumption prediction have been applied with varying levels of physics integration. RBF (Radial Basis Function) neural networks trained on QAR data have been used for phase-specific fuel prediction, achieving prediction errors of 5.73%, 3.36%, and 14.04% for takeoff/climb, cruise, and descent phases, respectively [4]. LSTM (Long Short-Term Memory)-based architectures with wavelet decomposition and metaheuristic hyperparameter optimization have reported R^2 values above 0.99 on specific aircraft types using range fuel data [5]. CEEMDAN (Complete Ensemble Empirical Mode Decomposition with Adaptive Noise)-decomposed QAR features combined with DMPSO (Dynamic Multidimensional Particle Swarm Optimisation)-optimized LSTM have been shown to reduce MAE (Mean Absolute Error) by over 40% compared to baseline neural networks in climb and cruise segments [6]. These approaches operate primarily on high-resolution onboard QAR data, which provides richer parameter sets than ADS-B but is not openly available. In an alternative data-driven formulation, Chati [9] developed Gaussian Process Regression models trained on flight data recorder records to predict instantaneous fuel flow and cumulative fuel burn, reporting up to 63% improvement in total airborne fuel burn prediction error relative to BADA, and additionally proposed a takeoff weight estimation method from ground-roll surveillance data. Choi et al. [10] trained decision tree and random forest regressors on operational, weight, and en-route weather variables for runway-to-runway fuel burn prediction, confirming that takeoff weight is a dominant predictor and that mean weather observations along the route provide sufficient meteorological representation for the regression task.

Concurrent work on the PRC (Performance Review Commission) Data Challenge 2025 [16] applied LightGBM [17] regression to ADS-B trajectories enriched with ERA5 wind data, OpenAP-derived mass and fuel-flow estimates, and statistical embeddings of trajectory variables. That study found that training on fuel flow rather than directly on fuel consumption improved predictive accuracy, and that heuristic mass interpolation between takeoff and landing weight outperformed recursive OpenAP-based mass integration, achieving RMSE (Root Mean Square Error) below 200 kg on the challenge ranking set.

The present work combines elements of both paradigms. It employs physics-informed feature engineering—including total energy model mass estimation [18, 19], OpenAP-derived fuel flow [12], and ERA5 wind-corrected true airspeed—within a gradient boosting regression framework (XGBoost [20], LightGBM [17]), with hyperparameters tuned via Optuna [21], operating on segment-level ADS-B and ACARS trajectory data from the PRC Data Challenge 2025 [22].

3. METHODOLOGY

3.1 Overview of the pipeline

The end-to-end workflow can be understood as five successive stages: Data ingestion and validation, including core challenge tables and external enrichment sources. Trajectory-based segment construction, using time-bounded subsets of high-frequency positional data. Meteorological augmentation, from both daily airport station data and ERA5 reanalysis. Physics-informed feature derivation, including true airspeed, flight phase, estimated mass, and fuel-flow surrogates. Model training and results generation, using optimized gradient boosting regressors.

3.2 Data sources and initial preparation

The data used in this study were downloaded from the repository provided in the PRC Data Challenge [22]. These data are in the format of a training dataset – that is, they contain a target variable: fuel burned between the start and end of specific flight segments, measured in kg. The flight dataset includes information on 11,037 flights across Europe between April and September 2025. The fuel-burned dataset includes data on flight segments and the fuel burned during those segments. Additionally, there are 3.1 GB of trajectory data collected from ADS-B and ACARS sources, stored in Parquet file format. The last dataset provides information on airport coordinates. For data processing, we employed the 'duckdb'[23] and 'pandas'[24] libraries in Python.

Flight data is structured as in Table 1.

Table 1. Flight Data Parameters Description

Column	Description	Format/Source
flight_date	Date of the flight	[YYYY-MM-DD]
aircraft_type	ICAO aircraft designator	-
takeoff	Takeoff time	[UTC]
landed	Landing time	[UTC]
origin_icao	Origin airport code	ICAO
destination_icao	Destination airport code	ICAO
destination_name	Name of the destination airport	-
flight_id	Unique flight identifier	-

Fuel-burned data is structured as in Table 2.

Table 2. Fuel-burned Data Parameters

Column	Description	Note/Unit
idx	row index for each record	-
flight_id	an identifier for the flight	-
start	a time instant	[UTC]
end	a time instant	[UTC]
fuel_kg	fuel burnt between start and end	[kg]

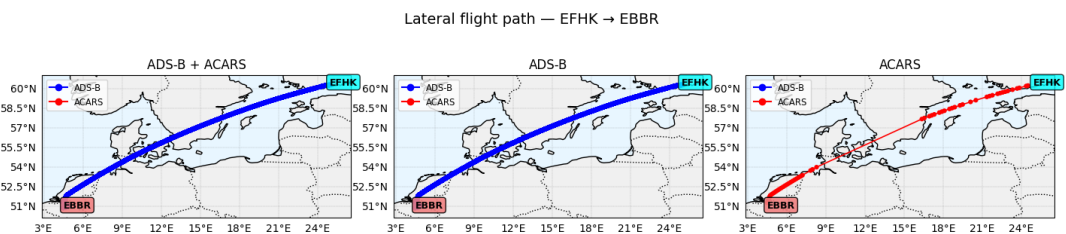
The 'fuel_kg' value in the last row represents our target variable in the machine learning process. The 'flight_id' is a helper value included in the dataset to uniquely identify each specific flight. It serves as an identifier and is not used as an input feature for prediction.

Trajectory data is structured as in Table 3. The trajectory dataset exhibits a source imbalance of

Table 3. Trajectory Data Parameters

Column	Description
timestamp	timestamp for the position report [UTC]
flight_id	an identifier for the flight
typecode	the ICAO aircraft type
latitude	latitude in decimal degrees in [-90, 90] range
longitude	longitude in decimal degrees in [-180, 180] range
altitude	altitude in feet
groundspeed	groundspeed in knots
track	track angle in decimal degrees
vertical_rate	vertical rate of climb/descent [ft/min]
mach	Mach number (from source = acars)
TAS	True Air Speed in knots (from source = acars)
CAS	Calibrated Air Speed in knots (from source = acars)
source	source of the position report (adsb or acars)

approximately 926:1 by record count, with 123,960,325 ADS-B observations versus 133,725 ACARS observations. ADS-B provides continuous positional coverage across all segments but carries only groundspeed, track angle, altitude, and vertical rate. ACARS records are sparse in coverage but supply direct onboard avionics measurements of TAS (True Air Speed), CAS (Calibrated Air Speed), and Mach number, which are inputs to the airspeed reconstruction fallback sequence described in Section 3.4. Because indiscriminate pooling of these sources would allow the numerically dominant ADS-B stream to suppress ACARS-derived kinematic parameters during aggregation, segment statistics are computed separately per source. The subsequent collapse to a single row per ('flight_id', 'segment_idx') pair applies 'source_percentage' – the fraction of records attributed to each source within the segment – as an aggregation weight, ensuring that ACARS-derived values contribute in proportion to their actual coverage rather than being diluted by ADS-B volume. Figure 1 shows ACARS trajectory in red and the ADS-B one in blue, for sample flight 'prc789748784'.

**Figure 1.** Plot of sample flight

Airports data is structured as follows:

An explicit data quality procedure addresses missing airport elevations. Using an alternative airport source [25], we successfully recovered 16 of 219 missing airport elevations. While incomplete, this imputation step is operationally relevant because airport elevation can influence both trajectory interpretation and surface-level weather context.

For each aircraft type in the Flight Data, we added extra features using data from the FAA Aircraft

Table 4. Airports Data Parameters

Column	Description
icao	ICAO airport code
longitude	longitude in decimal degrees in [-180, 180] range
latitude	latitude in decimal degrees in [-90, 90] range
elevation	Elevation in feet [ft]

Characteristics Database [26]. Table 5 explains meaning of specific parameters.

We download daily surface observations from the Iowa Environmental Mesonet API (METAR/ASOS, METeorological Aerodrome Report, Automated Surface Observing System). The data are retrieved in UTC with explicit latitude and longitude coordinates. Used API URL was [27] (Table 6). This dataset is later merged with flights by date and airport, enabling origin and destination weather descriptors to be attached to each segment.

A second meteorological data stream is derived from the Copernicus CDS (Climate Data Store). Our code constructs a per-flight request list and retrieves ERA5 pressure-level GRIB (GRIdded Binary) files. Requests are generated based on the spatiotemporal frame from trajectory data analysis, as explained in Subsection 3.3. We used the Python 'cdsapi'[28] library to access the 'reanalysis-era5-pressure-levels' dataset.

The purpose of this data is not to provide weather as a contextual feature, but to support later physical calculations, especially wind-corrected airspeed estimation and flight phase-based performance inference.

3.3 Trajectory coverage and segment extraction

Detailed flight trajectories stored in the trajectory dataset form the basis of the segment-level representation. For each labeled segment, our code filters pointwise trajectory data using the segment's start and end timestamps. The resulting segment subset is used to compute spatial and temporal motion summaries and to define the geographic footprint of the segment for atmospheric retrieval from CDS in the following way. For each flight segment, we determined the time ranges, lat/lon ranges, and altitude ranges. We had to convert altitude (in meters) to pressure (in hPa) to make a correct request to the CDS API, which accepts 'pressure_level', 'year', 'month', 'day', 'time', and 'area' [lat_max, lon_min, lat_min, lon_max]. To convert altitude ranges to pressure levels, we used the barometric formula:

$$P = P_0 \left(1 - \frac{L \cdot h}{T_0} \right)^{\frac{g \cdot M}{R \cdot L}} \quad (1)$$

Where P is the pressure at altitude h (in hPa), $P_0 = 1013.25$ hPa (sea-level standard atmospheric pressure), $L = 0.0065$ K/m (temperature lapse rate in the troposphere), h is the altitude (in meters), $T_0 = 288.15$ K (sea-level standard temperature), $g = 9.80665$ m/s² (standard gravitational acceleration), $M = 0.0289644$ kg/mol (molar mass of dry air), $R = 8.3144598$ J/(mol·K) (universal gas constant).

Retrieved atmospheric fields (Table 7) are used as direct inputs to kinematic and energy-based calculations.

3.4 Source-aware segment statistics

Trajectory data may originate from multiple surveillance sources, notably ADS-B and ACARS. Instead of pooling these indiscriminately, our pipeline computes statistics separately by source and

Table 5. FAA Aircraft Characteristics Database Parameters

Column	Description
ICAO_Code	ICAO Aircraft Type Designator
FAA_Designator	FAA Aircraft Type Designator
Manufacturer	Aircraft Manufacturer Name
Model_FAA	FAA Aircraft Model Name
Model_BADA	Eurocontrol BADA Aircraft Model Name
Physical_Class_Engine	Engine Class
Num_Engines	Number of Engines
AAC	Aircraft Approach Category
AAC_minimum	Minimum Aircraft Approach Category
AAC_maximum	Maximum Aircraft Approach Category
ADG	Aircraft Design Group
TDG	Taxiway Design Group
Approach_Speed_knot	Approach Speed at MALW
Appr_Speed_min_knot	Min Approach Speed at ALW
Appr_Speed_max_knot	Max Approach Speed at MALW
Wingspan_ft_no_wing	Wingspan (without winglets)
Wingspan_ft_wing	Wingspan (with winglets)
Length_ft	Aircraft Full Length
Tail_Height_ft	Tail Height at OEW
Wheelbase_ft	Aircraft Wheelbase
Cockpit_to_Gear_ft	Cockpit to Main Gear Distance
Main_Gear_Width_ft	Aircraft Main Gear Width
MTOW_lb	Maximum Takeoff Gross Weight
MALW_lb	Max Allowable Landing Weight
Main_Gear_Config	Main Gears Configuration
ICAO_WTC	ICAO Weight Turbulence Category
Parking_Area_ft2	Min Parking Position Sizing (Area)
Class	Aircraft Class
FAA_Weight	FAA Weight Class
CWT	CWT (Consolidated Wake Turbulence) Category
1.5_Wake_Category	RECAT (Re-categorisation) 1.5 Wake Category
2_Wake_Cat_AppxA	RECAT 2 Wake Category (Appx A)
2_Wake_Cat_AppxB	RECAT 2 Wake Category (Appx B)
Rotor_Diameter_ft	Rotor Diameter
SRS	SRS (Same Runway Separation) Category
LAHSO	LAHSO (Land and Hold Short Operations) Group
FAA_Registry	Flag for FAA Registry
Registration_Count	Registration Count (2022)
Total_IFR_Ops	Total IFR (Instrument Flight Rules) Operations (2021-2022)

Table 6. METAR Data Parameters

Column	Description
station	Three or four character site identifier
valid	Timestamp of the observation (local time)
lon	Longitude in decimal degrees [-180, 180]
lat	Latitude in decimal degrees [-90, 90]
tmpf	Air Temperature [°F] at 2 meters
dwpf	Dew Point Temperature [°F] at 2 meters
relh	Relative Humidity [%]
drct	Wind Direction [deg] from true north
sknt	Wind Speed [knots]
p01i	One hour precipitation [inches]
alti	Pressure altimeter [inches]
mslp	Sea Level Pressure [millibar]
vsby	Visibility [miles]
gust	Wind Gust [knots]
skyc1–skyc4	Sky Level 1–4 Coverage
skyl1–skyl4	Sky Level 1–4 Altitude [feet]
wxcodes	Present Weather Codes (space separated)
feel	Apparent Temperature [°F]
ice_accretion_1hr	Ice Accretion over 1 Hour [inches]
ice_accretion_3hr	Ice Accretion over 3 Hours [inches]
ice_accretion_6hr	Ice Accretion over 6 Hours [inches]
peak_wind_gust	Peak Wind Gust [knots] (from METAR)
peak_wind_drct	Peak Wind Gust Direction [deg]
peak_wind_time	Peak Wind Gust Time
metar	Unprocessed reported observation in METAR format
snowdepth	Snow Depth [inches] (from METAR)

Table 7. ERA5 Pressure-Level Data Parameters

Column	Description
time	Timestamp of the observation [UTC]
isobariclnhPa	Pressure level in hPa
latitude	Latitude in decimal degrees
longitude	Longitude in decimal degrees
valid_time	Valid time of the observation [UTC]
z	Geopotential in m^2/s^2
t	Temperature in Kelvin [K]
u	U component of wind in m/s
v	V component of wind in m/s
w	Vertical velocity in Pa/s

later aggregates them in a controlled way.

For each segment-source pairing, the following statistics are computed:

- time since take-off, and until landed,
- segment duration,
- counts of ADS-B and ACARS observations,
- source share within the segment (source_percentage),
- mean and standard deviation of altitude, groundspeed, vertical rate and true airspeed,
- mean latitude and longitude,
- Haversine distance,
- three-dimensional distance.

The framework also attempts to reconstruct missing kinematic information through a structured fallback sequence. Missing true airspeed is filled, when possible, from:

- direct TAS values,
- calibrated airspeed (CAS),
- Mach number together with atmospheric temperature,
- groundspeed and track corrected by ERA5 wind components.

Conversely, missing groundspeed can be reconstructed from TAS, track, and wind. This step materially strengthens the feature representation by exploiting physical consistency across multiple observed or inferred variables.

In instances where TAS must be derived solely from CAS and neither static air temperature nor pressure altitude is available at the trajectory point level, CAS is used as a first-order proxy for TAS. This substitution is physically exact only at mean sea level under ISA (International Standard Atmosphere) standard conditions ($T_0 = 288.15$ K, $P_0 = 1013.25$ hPa). At operationally representative cruise levels (FL300–FL380, Flight Level 30,000 feet–Flight Level 38,000 feet), TAS exceeds CAS by approximately 30–35%, a divergence that grows monotonically with altitude due to decreasing air density and compressibility effects. The isentropic correction from CAS to TAS requires knowledge of ambient static pressure and temperature, neither of which was consistently available per trajectory point for the affected subset of records. The resulting underestimation in TAS propagates to the specific energy rate (Eq. 7) and, through the quadratic mass balance (Eq. 13), to the per-phase mass and fuel-flow estimates. The fraction of segments for which this fallback was invoked, and its quantitative effect on final RMSE, were not independently assessed.

True airspeed is calculated from Mach number using the speed of sound:

$$TAS = M \cdot a \quad (2)$$

$$a = \sqrt{\gamma \cdot R_{specific} \cdot T} \quad (3)$$

Where M is the Mach number (dimensionless), a is the speed of sound (m/s), $\gamma = 1.4$ is the adiabatic index for dry air, $R_{specific} = 287.05$ J/(kg·K) is the specific gas constant for dry air, and T is the static air temperature (K). The temperature T at a specific altitude is calculated from ERA5 data by

interpolation between provided data points. The geopotential z is converted to geometric height by dividing by $g_0 = 9.80665 \text{ m/s}^2$ (standard gravity).

True airspeed is calculated from groundspeed, track angle, and ERA5 wind components using vector subtraction:

$$\vec{V}_{TAS} = \vec{V}_{GS} - \vec{V}_{wind} \quad (4)$$

$$TAS = |\vec{V}_{TAS}| = \sqrt{u_{TAS}^2 + v_{TAS}^2} \quad (5)$$

Where $\vec{V}_{TAS} = (u_{TAS}, v_{TAS})$ is the true airspeed vector (m/s), $\vec{V}_{GS} = (u_{GS}, v_{GS})$ is the groundspeed vector (m/s), and $\vec{V}_{wind} = (u_{wind}, v_{wind})$ is the wind vector (m/s). The components $u_{GS} = GS_{m/s} \cdot \cos(\theta_{math})$ and $v_{GS} = GS_{m/s} \cdot \sin(\theta_{math})$ represent the East/North groundspeed components, where $\theta_{math} = 90^\circ - \text{track}_{deg}$ converts the meteorological convention ($0^\circ = \text{North}$) to the mathematical convention ($0^\circ = \text{East}$). The conversion from knots to m/s is given by $GS_{m/s} = \text{groundspeed}_{kts} \times 0.514444$.

Once segment-level source statistics are available, the code uses a special function to collapse multiple source-specific rows into a single row per `flight_id` and `segment_idx`. For selected motion-related variables, weighted averages are computed using `source_percentage` (the fraction of records in a segment originating from ADS-B or ACARS) as the weight. Other variables are preserved using first-value aggregation, while bookkeeping columns such as `acars_count`, `adsb_count`, and `source_percentage` are excluded from the final unified output. This fusion strategy is methodologically sound because it allows high-value sources to contribute proportionally while still generating a compact segment table suitable for machine learning.

METAR/ASOS data points are joined to flights by airport and date, allowing origin and destination meteorological conditions to be attached to each flight segment.

3.5 Physics-informed mass and fuel-flow features

The next enrichment stage identifies operational flight phases from time-ordered trajectories using Python ‘traffic’[29] library and included method ‘phases’. The model uses elapsed time, altitude, groundspeed, and vertical rate to assign labels such as climb (CL), descent (DE), and cruise (CR). The workflow first checks whether a segment contains climb and descent portions and later expands the logic to store separate subsets for climb, descent, and cruise trajectories. This shift from aggregate segment summaries to explicit phase segmentation is a major methodological contribution, because fuel burn is inherently phase-dependent. One of the most domain-informed components of the workflow is implemented in the following stage. The code uses OpenAP drag, thrust, and fuel-flow models, together with ERA5-derived temperature and wind data, to estimate aircraft mass and phase-specific fuel-flow summaries. The procedure includes:

- interpolating weather profiles to the aircraft location, altitude, and time,
- computing drag coefficients from an OpenAP aircraft model,
- estimating thrust for climb, descent, and cruise,
- computing an energy-rate balance from altitude and airspeed changes,
- solving for a least-squares mass estimate,
- deriving fuel flow at estimated thrust.

The resulting segment-level features include:

- estimated mass for climb phase,

- estimated mass for descend phase,
- fuel flow for climb phase,
- fuel flow for descend phase,
- fuel flow for cruise phase.

These variables represent an important bridge between raw surveillance trajectories and the underlying aircraft performance mechanisms that determine fuel burn. Before any kinematic and energy-based calculations, we filtered the trajectory data because it usually contains noise. Figure 2 shows the results of the filtering process for sample segment. We use the total energy model to estimate aircraft mass during the climb and descent phases:

$$\frac{T - D}{m} \cdot TAS - \dot{E} = 0 \quad (6)$$

where $\dot{E}$ ($\text{J} \cdot \text{kg}^{-1} \cdot \text{s}^{-1}$) is the specific energy rate, defined as the total time derivative of the sum of specific kinetic and specific potential energy:

$$\dot{E} = \frac{1}{2} \cdot \frac{d(TAS^2)}{dt} + g_0 \cdot \frac{dh}{dt} \quad (7)$$

The first term captures the rate of change of longitudinal kinetic energy per unit mass due to along-track acceleration; the second term captures the rate of change of potential energy per unit mass due to altitude variation. T is Thrust (N), D is Drag (N), m is Aircraft mass (kg), TAS is True airspeed (m/s), h is geometric altitude (m), and $g_0 = 9.80665 \text{ m/s}^2$ is the standard gravitational acceleration. By expressing the energy balance per unit mass (Eq. 6), thrust, drag, and kinematics are jointly constrained such that m is the sole unknown resolvable from observable trajectory quantities (TAS , $dTAS/dt$, dh/dt) and model-derived T and D values.

Aerodynamic drag is approximated by a second-order mass-dependent model. Under the quasi-steady flight assumption ($L \approx mg$), the lift coefficient scales linearly with m , and induced drag is proportional to m^2 . Total drag is expressed as:

$$D = c_0 + c_1 \cdot m^2 \quad (8)$$

where c_0 represents the zero-lift (parasite) drag component and c_1 is an effective induced drag coefficient that absorbs the proportionality between induced drag and the square of lift-equivalent mass loading. Both coefficients are implicit functions of dynamic pressure.

$$q = \frac{1}{2} \rho TAS^2 \quad (9)$$

and are therefore altitude- and airspeed-dependent. In this implementation, c_0 and c_1 are calibrated per segment at a single reference mass point $m_{ref} = 10,000 \text{ kg}$ using `openap.drag.Drag.clean()` evaluated at the segment's mean TAS and pressure altitude:

$$c_0 = D(m = 0), \quad c_1 = \frac{D(m_{ref}) - D(0)}{m_{ref}^2} \quad (10)$$

This per-segment evaluation partially accounts for the variation of q across flight levels. However, both coefficients (c_0, c_1) are treated as locally constant over the segment duration, which constitutes a valid approximation for short, altitude-stable segments but may introduce systematic drag estimation error in segments that span large altitude or airspeed ranges.

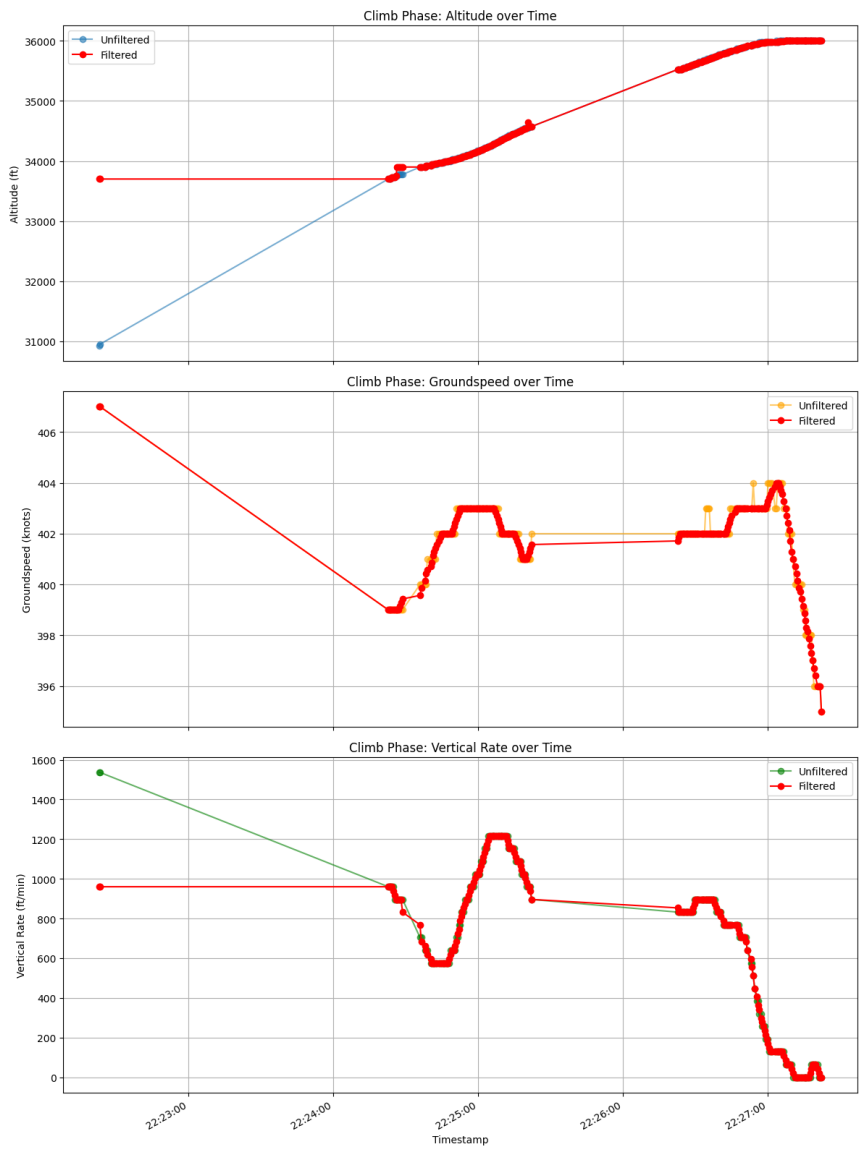


Figure 2. Filtered results for sample segment

To compute maximum thrust across flight phases, we used OpenAP's 'thrust.Thrust' model (with phase-specific methods):

$$T = \begin{cases} T_{climb}(TAS, h, ROC) & \text{if phase = CL} \\ T_{descent}(TAS, h, ROC) & \text{if phase = DE} \\ T_{cruise}(TAS, h) & \text{if phase = CR} \end{cases} \quad (11)$$

Where ROC is rate-of-climb (ft/min), i.e. vertical rate.

To estimate mass at each trajectory point, we solve the total energy balance equation using a physics-constrained quadratic formulation:

$$\frac{(T - D) \cdot TAS}{m} = \dot{e} \quad (12)$$

With $D = c_0 + c_1 m^2$, we rearrange to the standard quadratic:

$$-c_1 \cdot TAS \cdot m^2 + (T - c_0) \cdot TAS - \dot{e} \cdot m = 0 \quad (13)$$

Or: $am^2 + bm + c = 0$, where: $a = -c_1 \cdot TAS$, $b = -\dot{e}$, $c = (T - c_0) \cdot TAS$.

Equation (13) is a quadratic polynomial in m of the form $am^2 + bm + c = 0$, where $a = -c_1 \cdot TAS$, $b = -\dot{e}$, and $c = (T - c_0) \cdot TAS$. We solved this using the quadratic formula and selected the positive, physically realistic root within operational mass bounds for the aircraft type – with the OpenAP-tabulated Operating Empty Weight (OEW) as the lower bound and the Maximum Takeoff Weight (MTOW) as the upper bound.

For this task we used 'minimize_scalar' from 'scipy.optimize'[30] – it finds the single mass value that minimizes the sum of squared errors across all samples in the segment. If the optimization process fails to find a solution (for example, if the objective function cannot be minimized within the given bounds), the mass estimate for that trajectory point is flagged as invalid and excluded from segment-level aggregation.

Mass estimates are computed for all trajectory points classified as climb (CL) or descent (DE) phase; cruise-phase points (CR) are excluded because the near-zero vertical rate renders the potential energy term in Eq. (7) negligible, making the energy balance ill-conditioned for mass inference from altitude kinematics alone.

3.6 Final feature assembly and cleaning

After phase-aware enrichment, we combined segment summaries with airport, aircraft, and weather information. This final modeling table then underwent several cleaning operations, including:

- dropping selected identifiers (columns not useful for model training),
- removing columns that were entirely missing or contained >50% NaN values,
- processing datetime variables (extracting hour, minute, day-of-week, month, and year from each datetime column, then removing redundant originals),
- dropping single-value columns,
- casting object-type predictors to categorical dtype for regressors.

The target variable was `fuel_kg`, and the final feature matrix had 73 features and 131,530 rows.

3.7 Predictive modeling

We chose XGBoost (eXtreme Gradient Boosting) for our numerical prediction task due to its established performance in regression problems. The library leverages gradient-boosted decision trees, making it particularly effective for our dataset. Hyperparameter tuning was conducted using Optuna across number of trials with the TPESampler, which applies the Tree-structured Parzen Estimator for efficient search. Table 8 details the optimized parameters used in the XGBoost model.

Table 8. XGBoost Hyper-paramaters

Parameter Name	Description	Value / Range
n_estimators	Number of trees	10,000
learning_rate	Tree weights shrinking	0.001 – 0.4 (log)
max_depth	Depth of trees	3 – 12
gamma	Regularization parameter	0 – 10
min_child_weight	Minimum sum of instance weight	1 – 20
subsample	Fraction of samples for training	0.5 – 1.0
colsample_bytree	Subsample ratio of columns	0.5 – 1.0
colsample_bylevel	Column subsample ratio per level	0.5 – 1.0
colsample_bynode	Column subsample ratio per node	0.5 – 1.0
reg_alpha	L1 regularization	0.01 – 10 (log)
reg_lambda	L2 regularization	0.01 – 10 (log)
max_delta_step	Maximum delta step per tree update	0 – 10

We also implemented LightGBM for the same regression task, leveraging its gradient-boosted decision trees optimized for high-dimensional tabular data. Hyperparameters were tuned using the same Optuna framework (TPE sampler), adapting LightGBM-specific parameters (Table 9).

Table 9. LightGBM Hyper-paramaters

Parameter Name	Description	Value / Range
n_estimators	Number of trees	50,000
learning_rate	Tree weights shrinking	0.001 – 0.3 (log)
num_leaves	Maximum leaves per tree	20 – 100
max_depth	Maximum tree depth	3 – 12
min_child_samples	Minimum samples per leaf	1 – 20
subsample	Row sampling fraction	0.5 – 1.0
colsample_bytree	Column sampling per tree	0.5 – 1.0
reg_alpha	L1 regularization	0.01 – 10 (log)
reg_lambda	L2 regularization	0.01 – 10 (log)
min_split_gain	Min loss reduction for split	0 – 10
bagging_fraction	Fraction of data for bagging	0.5 – 1.0
bagging_freq	Bagging frequency	1 – 7

Both XGBoost and LightGBM models were trained using K-fold cross-validation to ensure robust generalization across the diverse flight in our dataset. This approach systematically partitioned the

segment-level feature matrix into K folds, training on K-1 folds while evaluating on the held-out fold, repeated across all permutations.

4. RESULTS

On the provided training dataset, XGBoost achieved an RMSE of 232.76 kg, while LightGBM reached 243.91 kg (K-fold cross-validation). Our best submitted model for the PRC 2025 challenge leaderboard scored 227.32 kg RMSE on the final blind test set. The final submission dataset was a holdout set provided without target values, enabling fair, unbiased evaluation of all competing models on identical test conditions. Figures 3 and 4 show feature importance for our best submitted model. Figure 3 displays LightGBM gain importance (the contribution of each feature to loss reduction). Figure 4 shows LightGBM split importance (the number of times a feature was used to split a node across all trees).

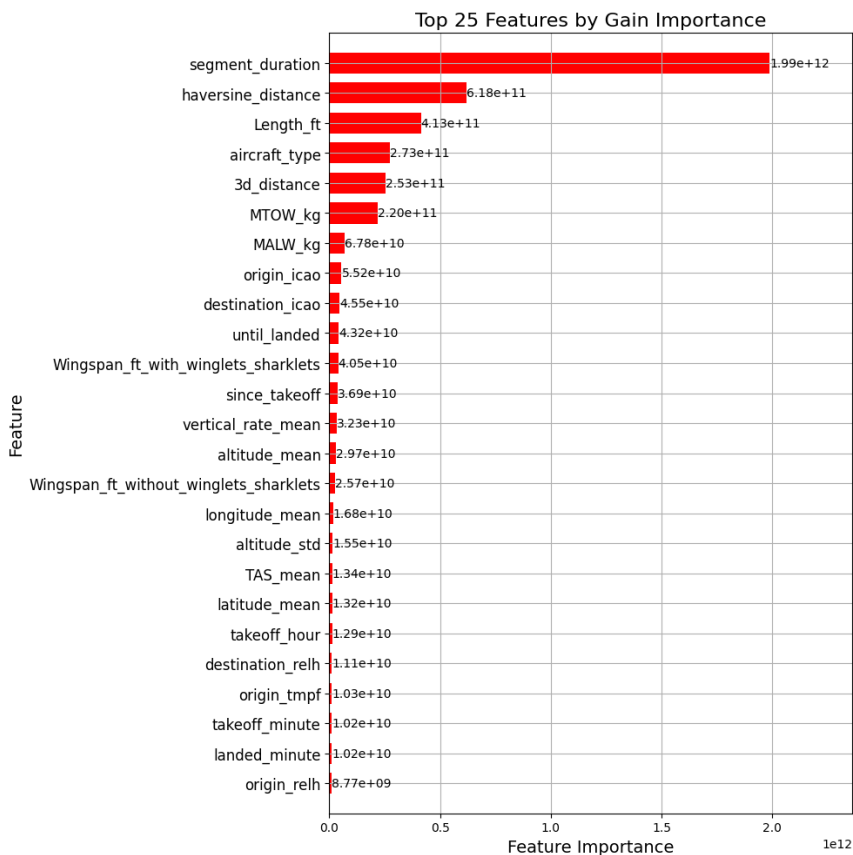


Figure 3. LightGBM feature importance (Gain)

To visually evaluate and demonstrate the performance of our best-performing model, we generated out-of-fold (OOF) predictions across the entire dataset using a 10-fold cross-validation framework. For each iteration, the best-performing model configuration was trained on the training folds and evaluated on the corresponding held-out fold. Because the final test set lacked target variables, the OOF RMSE serves as our proxy for expected test performance, ensuring that each prediction was generated by a model that had not previously encountered that specific sample. Figure 5 (left)

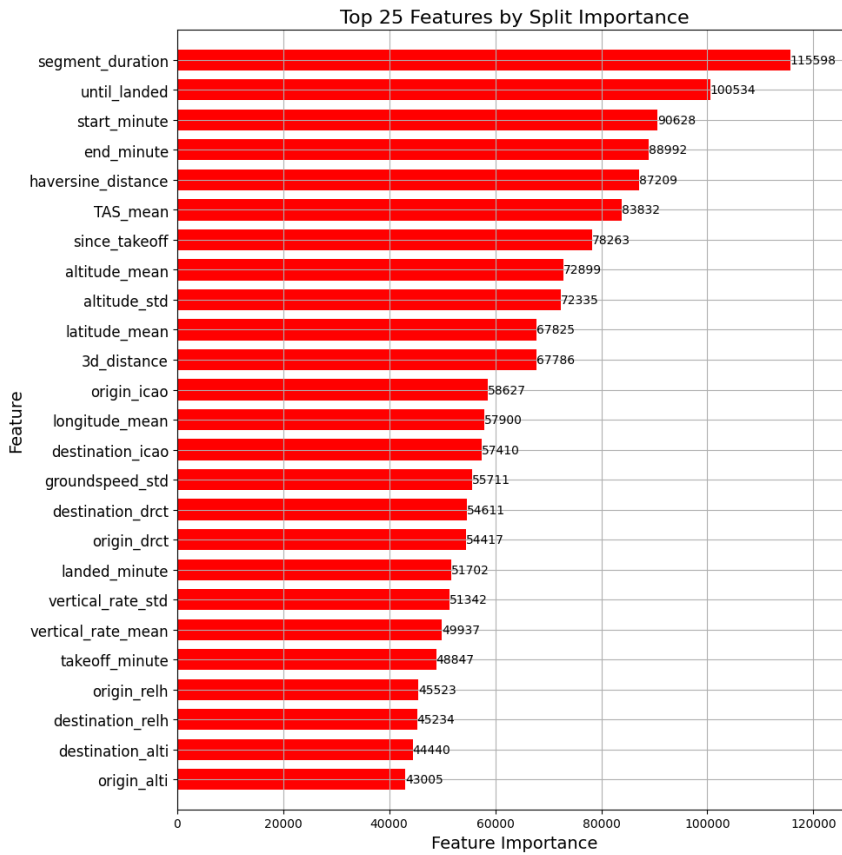


Figure 4. LightGBM feature importance (Split)

displays the parity plot comparing the actual fuel consumption against the OOF predictions, where the dashed line represents perfect prediction (the identity line); the model achieved an OOF RMSE of 203.3 kg. Figure 5 (right) illustrates the distribution of the residuals (true - OOF prediction) via a histogram.

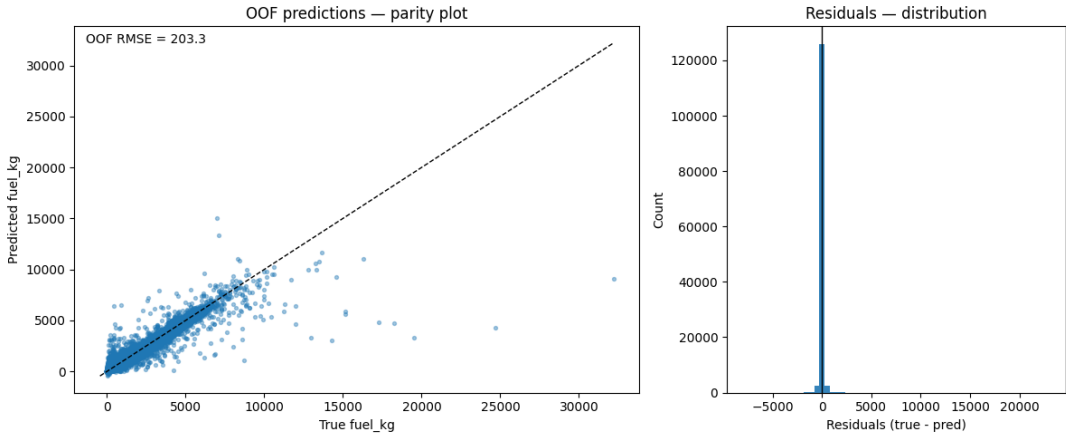


Figure 5. Out-of-fold error analysis for the LightGBM model

5. CONCLUSION AND FUTURE WORK

This work applied gradient boosting regression — XGBoost and LightGBM, both tuned via Optuna TPE search — to a 73-feature, segment-level representation of 131,530 labeled fuel burn records from the PRC Data Challenge 2025. The feature space was constructed to encode physically motivated quantities: ERA5 wind-corrected true airspeed, OpenAP-derived drag and thrust coefficients, total energy model mass estimates for climb and descent phases, and phase-specific fuel-flow summaries assigned by the traffic library phase detector. Source-aware aggregation of ADS-B and ACARS trajectory statistics was performed using source-percentage-weighted averages. XGBoost achieved a K-fold cross-validation RMSE of 232.76 kg; LightGBM achieved 243.91 kg under the same evaluation protocol. The best submission to the challenge blind test set scored 227.32 kg RMSE. The 4.9 kg gap between the XGBoost cross-validation estimate and the blind test result is consistent with standard generalization variance at this dataset scale, but no formal confidence interval over folds was reported, and the gap should not be interpreted as evidence of positive out-of-distribution behavior.

Several methodological limitations apply. First, the fallback procedure that equates calibrated airspeed to true airspeed — applicable only under sea-level ISA conditions — introduces a systematic positive bias in all downstream aerodynamic calculations at cruise altitude, where TAS typically exceeds CAS by 30–35%. The magnitude of this bias on the final RMSE was not quantified. Second, no coverage analysis was reported for missing trajectory segments; the fraction of records for which physics-informed features could not be computed and were therefore imputed or dropped is unknown. Third, the total energy model mass estimates depend on OpenAP drag and thrust tabulations calibrated to median aircraft performance; aircraft-to-aircraft variation within a type designator, engine wear, and payload distribution are not modelled, and no uncertainty estimates accompany the mass outputs. Fourth, the evaluation metric — RMSE in kg — does not distinguish whether residual error is distributed uniformly across segment durations and aircraft types or concentrated in specific operational conditions such as long-haul cruise segments or heavy wide-body types. Feature importance analysis or Shapley value decomposition, which the current implementation omits, would be

required to assess which sources of variance dominate.

Directions for further development include: explicit modeling of the TAS–CAS conversion using ISA compressibility and density ratio corrections; quantitative assessment of trajectory coverage and its relationship to prediction error; ensemble combination of XGBoost and LightGBM outputs with empirical calibration; and evaluation of prediction error stratified by aircraft type, segment duration, and data source proportion, to determine whether the current pipeline generalises uniformly or carries structural biases against specific flight categories.

Author contributions

- Rade Kačar: Conceptualization, Data Curation, Formal Analysis, Funding Acquisition, Investigation, Methodology, Project Administration, Resources, Software, Supervision, Validation, Visualization, Writing (Original Draft), Writing (Review and Editing)
- Darko Čulibrk: Conceptualization, Data Curation, Formal Analysis, Funding Acquisition, Investigation, Methodology, Project Administration, Resources, Software, Supervision, Validation, Visualization, Writing (Original Draft), Writing (Review and Editing)

Open data statement

The source code for the preprocessing pipeline, feature engineering, and model training is available at <https://github.com/radekacar/kindly-garden> and <https://doi.org/10.5281/zenodo.19316338>.

Reproducibility statement

The source code for the preprocessing pipeline, feature engineering, and model training is available at <https://github.com/radekacar/kindly-garden> and <https://doi.org/10.5281/zenodo.19316338>.

References

- [1] Angela Nuic. *User Manual for the Base of Aircraft Data (BADA) Revision 3.13*. Tech. rep. EEC Technical/Scientific Report No. 15/04/02-43. EUROCONTROL Experimental Centre, 2015.
- [2] Angela Nuic, Damir Poles, and Vincent Mouillet. “BADA: An advanced aircraft performance model for present and future ATM systems”. In: *International Journal of Adaptive Control and Signal Processing* 24.10 (2010), pp. 850–866. DOI: 10.1002/acs.1176.
- [3] R. Dalmau, X. Prats, A. Ramonjoan, and S. Soley. “Estimating fuel consumption from radar tracks: a validation exercise using FDR and radar tracks from descent trajectories”. In: *CEAS Aeronautical Journal* 11.2 (2020), pp. 355–365. DOI: 10.1007/s13272-020-00441-2.
- [4] Y. Zhao, Z. Wang, X. Wang, Y. Song, and Y. Han. “Data driven fuel consumption prediction model for green aviation using radial basis function neural network”. In: *Scientific Reports* 15.1 (2025), p. 26275. DOI: 10.1038/s41598-025-11941-8.
- [5] Weizhen Tang, Jie Dai, and Yuntai Li. “Aircraft range fuel prediction study based on WPD with IAPO optimized BiLSTM–KAN model”. In: *Scientific Reports* 15.1 (2025), p. 12231. DOI: 10.1038/s41598-025-97264-0.
- [6] J. Xiong, C. Zou, Y. Wan, Y. Sun, and G. Yu. “Fuel Consumption Prediction for Full Flight Phases Toward Sustainable Aviation: A DMPSO–LSTM Model Using Quick Access Recorder (QAR) Data”. In: *Sustainability* 17.8 (2025), p. 3358. DOI: 10.3390/su17083358.
- [7] Karolina Krajček Nikolić, Petar Papoči, Dario Nikolić, and Bruno Antulov-Fantulin. “Fuel Burn Method Assessment Using Automatic Dependent Surveillance–Broadcast and European Re-analysis Data: Limited Flight Sample Analysis”. In: *Aerospace* 11.2 (2024), p. 154. DOI: 10.3390/aerospace11020154.

- [8] M. Ahearn et al. *Aviation Environmental Design Tool (AEDT) Technical Manual, Version 2b, Service Pack 3*. Tech. rep. DOT-VNTSC-FAA-16-11. U.S. Department of Transportation Volpe National Transportation Systems Center, 2016.
- [9] Y. S. Chati. “Statistical modeling of aircraft engine fuel burn”. PhD thesis. Cambridge, MA: Massachusetts Institute of Technology, 2018.
- [10] H. Choi, A. Behere, J. Bhanpato, M. Kirby, and D. N. Mavris. “Prediction of Fuel Burn for Runway-to-Runway Commercial Flight Operations with Machine Learning”. In: *AIAA AVIATION FORUM AND ASCEND 2025*. AIAA 2025-3519. July 2025. doi: 10.2514/6.2025-3519.
- [11] Gano Broto Chatterji. “Fuel Burn Estimation Using Real Track Data”. In: *11th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*. 2011. doi: 10.2514/6.2011-6881.
- [12] J. Sun, J. M. Hoekstra, and J. Ellerbroek. “OpenAP: An Open-Source Aircraft Performance Model for Air Transportation Studies and Simulations”. In: *Aerospace 7.8* (2020), p. 104. doi: 10.3390/aerospace7080104.
- [13] J. Sun. “Open Aircraft Performance Modeling: Based on an Analysis of Aircraft Surveillance Data”. PhD thesis. Delft, Netherlands: Delft University of Technology, 2019. doi: 10.4233/uuiid:af94d535-1853-4a6c-8b3f-77c98a52346a.
- [14] A. Filippone, B. Parkes, N. Bojdo, and T. Kelly. “Prediction of aircraft engine emissions using ADS-B flight data”. In: *The Aeronautical Journal* 125.1288 (2021), pp. 988–1012. doi: 10.1017/aer.2021.2.
- [15] A. Filippone, N. Bojdo, S. Mehta, and B. Parkes. “Using the OpenSky ADS-B Data to Estimate Aircraft Emissions”. In: *Engineering Proceedings* 13.1 (2021), p. 11. doi: 10.3390/engproc2021013011.
- [16] Team Resourceful Quiver. *Fuel Consumption Estimation*. PRC Data Challenge 2025 submission report. Delft University of Technology, 2025. URL: https://github.com/PRC-Data-Challenge-2025/resourceful-quiver/blob/main/PRC_2025_report.pdf (visited on 08/25/2026).
- [17] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. “LightGBM: A Highly Efficient Gradient Boosting Decision Tree”. In: *Advances in Neural Information Processing Systems (NeurIPS)*. Vol. 30. 2017.
- [18] R. Alligier, D. Gianazza, and N. Durand. “Energy rate prediction using an equivalent thrust setting profile”. In: *International Conference on Research in Air Transportation (ICRAT)*. May 2012.
- [19] Richard Alligier, David Gianazza, and Nicolas Durand. “Ground-based estimation of the aircraft mass, adaptive vs. least squares method”. In: *10th USA/Europe Air Traffic Management Research and Development Seminar (ATM 2013)*. June 2013.
- [20] Tianqi Chen and Carlos Guestrin. “XGBoost: A Scalable Tree Boosting System”. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2016, pp. 785–794. doi: 10.1145/2939672.2939785.
- [21] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. “Optuna: A Next-generation Hyperparameter Optimization Framework”. In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 2019, pp. 2623–2631. doi: 10.1145/3292500.3330701.
- [22] Performance Review Commission. *PRC Data Challenge 2025*. EUROCONTROL. 2025. URL: <https://ansperformance.eu/study/data-challenge/dc2025/> (visited on 10/11/2025).
- [23] Mark Raasveldt and Hannes Mühleisen. “DuckDB: an Embeddable Analytical Database”. In: *Proceedings of the 2019 International Conference on Management of Data*. 2019, pp. 1981–1984. doi: 10.1145/3299869.3320212.
- [24] Wes McKinney. “Data Structures for Statistical Computing in Python”. In: *Proceedings of the 9th Python in Science Conference*. Ed. by Stéfan van der Walt and Jarrod Millman. 2010, pp. 56–61. doi: 10.25080/Majora-92bf1922-00a.

- [25] David Megginson. *OurAirports Airport Data (airports.csv)*. OurAirports, 2025. URL: <https://ourairports.com/data/> (visited on 10/11/2025).
- [26] Federal Aviation Administration. *Aircraft Characteristics Database (October 2023)*. U.S. Department of Transportation. Oct. 2023. URL: https://www.faa.gov/airports/engineering/aircraft_char_database (visited on 10/11/2025).
- [27] Iowa Environmental Mesonet. *Automated Surface Observing System (ASOS) Data Download API*. Iowa State University Department of Agronomy. 2025. URL: <https://mesonet.agron.iastate.edu/cgi-bin/request/asos.py> (visited on 10/11/2025).
- [28] European Centre for Medium-Range Weather Forecasts. *cdsapi: Python API to access the Copernicus Climate Data Store*. Version 0.7.7. Sept. 2025. URL: <https://github.com/ecmwf/cdsapi>.
- [29] Xavier Olive. “traffic, a toolbox for processing and analysing air traffic data”. In: *Journal of Open Source Software* 4.39 (2019), p. 1518. DOI: 10.21105/joss.01518. URL: <https://doi.org/10.21105/joss.01518>.
- [30] Pauli Virtanen et al. “SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python”. In: *Nature Methods* 17.3 (2020), pp. 261–272. DOI: 10.1038/s41592-019-0686-2. URL: <https://doi.org/10.1038/s41592-019-0686-2>.