On the generalization properties of deep learning for aircraft fuel flow estimation models

Jarry, Gabriel; Dalmau, Ramon; Very, Philippe; Sun, Junzi

DOI
10.1016/j.trc.2025.105143

Publication date
2025

Document Version
Final published version

Published in
Transportation Research Part C: Emerging Technologies

Transportation Research Part C 176 (2025) 105143

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

On the generalization properties of deep learning for aircraft fuel flow estimation models

Gabriel Jarry [a], Ramon Dalmau [a], Philippe Very [a], Junzi Sun [b,*]

[a] EUROCONTROL, Brétigny-Sur-Orge, France
[b] Faculty of Aerospace Engineering, Delft University of Technology, Kluyverweg 1, 2629 HS Delft, The Netherlands

ARTICLE INFO

Keywords:
Fuel flow estimation
Aviation sustainability
Machine learning
Neural network
Domain generalization

ABSTRACT

Accurately estimating aircraft fuel flow is critical for evaluating new procedures, designing next-generation aircraft, and monitoring the environmental impact of current aviation practices. This paper investigates the generalization capabilities of deep learning models for fuel flow prediction, focusing on their performance with aircraft types not included in the training data. We propose a novel methodology that combines neural network architectures with domain generalization techniques to improve robustness and reliability across different aircraft types. Using a comprehensive dataset of 101 aircraft types, split into training (64 types) and generalization (37 types) sets with each type represented by 1,000 flights, we introduce a pseudo-distance metric to quantify aircraft type similarity and explore sampling strategies to improve model performance in data-limited regions. Our findings show that for unseen aircraft types, especially with noise regularization, the model outperforms baselines such as corrected proxy estimates. This study demonstrates the potential of blending domain-specific insights with advanced machine learning techniques to develop scalable, accurate, and generalizable fuel flow estimation models.

1. Introduction

The aviation industry is a notable contributor to global CO2 emissions, responsible for approximately 2.5% of the total – a share expected to increase as air travel continues to expand Gössling and Humpe (2020). In addition to CO2, aviation generates non-CO2 effects, such as contrail formation, which further contribute to global warming (Lee et al., 2021). As awareness of these environmental impacts grows, the industry has set ambitious goals to reduce its carbon footprint. International initiatives, such as the U.S. NextGen program and the Single European Sky Air Traffic Management Research (SESAR) Joint Undertaking, aim to significantly reduce aviation's environmental impact by 2035 through a variety of technological advancements and procedural innovations (SESAR, 2015).

Achieving these goals necessitates a multi-faceted approach, including advancements in engine technology, more aerodynamically efficient wing designs, and optimized air traffic management (ATM) procedures. For example, operational practices like continuous (cruise) climb operations (CCOs) and continuous descent operations (CDOs) can significantly cut fuel consumption and emissions (Dalmau and Prats, 2015; Clarke et al., 2004). However, assessing the environmental benefits of such procedures depends on accurate and reliable fuel consumption models, which are also crucial for monitoring the environmental impact of current operations.

* Corresponding author.
E-mail addresses: gabriel.jarry@eurocontrol.int (G. Jarry), ramon.dalmau@eurocontrol.int (R. Dalmau), philippe.very@eurocontrol.int (P. Very), j.sun-1@tudelft.nl (J. Sun).

https://doi.org/10.1016/j.trc.2025.105143
Received 11 November 2024; Received in revised form 31 March 2025; Accepted 16 April 2025
Available online 5 May 2025

Numerous models have been developed to estimate fuel consumption, each employing different methodologies and data sources. Many rely on parametric models derived from basic physical equations or empirical data analysis (Nuic et al., 2010; Poll and Schumann, 2021b), others are using advanced machine learning techniques such as neural networks (Uzun et al., 2021). The data used to develop these models range from widely accessible sources like automatic dependent surveillance-broadcast (ADS-B) (Sun, 2022), to more detailed proprietary information from manufacturers (Nuic et al., 2010) or from Quick Access Recorder data (QAR) (Jarry et al., 2024a). While these models are valuable, they come with limitations. Parametric models, for instance, often oversimplify the complexities of fuel consumption, leading to potential inaccuracies in certain operational scenarios. For example, some models may provide excellent accuracy during the cruise phase, but not during the descent phase. Moreover, the simplicity of existing models typically necessitates the use of a separate model for each aircraft type and the use of proxy aircraft method when the aircraft type is not available. This fragmentation limits their applicability, especially in cases where data availability is sparse. If separate models are trained for each aircraft type, the data available for each may be insufficient to achieve high accuracy. Additionally, data for some aircraft types might be unavailable or only available for certain flight phases, such as cruise.

In response to these challenges, we extended the idea of the development of a generic and unique fuel consumption model (Jarry et al., 2024a; Chati and Balakrishnan, 2016) that is conditioned on specific characteristics. This approach offers several key advantages over traditional models. First, by conditioning on aircraft characteristics, a single model can be applied across multiple aircraft types, eliminating the need to develop and maintain a separate model for each type. Second, the model can be trained on a larger and more diverse dataset, encompassing various aircraft types and operational conditions, thereby enhancing its robustness and accuracy. Third, and perhaps most importantly, this model has the potential to generalize across different scenarios, including those not represented in the training data. This capability, known as domain generalization in machine learning (Zhou et al., 2022), makes it a powerful tool for what-if analyses and for forecasting fuel consumption under new or unforeseen conditions, such as new aircraft designs or operational procedures.

This paper presents this model and, most importantly, empirically evaluates its domain generalization properties. Ideally, such a model should be developed using QAR data (Jarry et al., 2024a). However, to overcome the lack of fleet representativeness in the available QAR data, the BADA 4.2.1 model is used as a ground truth for fuel flow. The authors acknowledge that this assumption is essential and has further implications that are discussed in Section 6. However, this is the most comprehensive aircraft performance model available, which we can use to study the generalization property in our research.

2. State of the art

This section presents the state of the art in two key areas. The first subsection reviews the latest advancements in fuel estimation techniques, while the second subsection explores the current approaches to domain generalization.

2.1. Fuel estimation

Accurately estimating aircraft fuel consumption remains a significant challenge in ATM. To address this, researchers have developed various modeling techniques, ranging from traditional physical models to advanced neural networks and machine learning approaches. This subsection highlights the most relevant works across these different methodologies.

2.1.1. Physical models

EUROCONTROL's Base of Aircraft Data (BADA) has been a widely recognized aircraft performance model that provides a foundational framework for simulating aircraft trajectories using total energy modeling. BADA offers comprehensive models for thrust, drag, and fuel consumption across a wide range of aircraft types. For instance, BADA 3 currently covers 97% of ECAC 2023 IFR flights, while BADA 4.2 covers only 73%. The forthcoming BADA 4.3 is expected to extend this coverage to 84.5% (Nuic and Mouillet, 2016; Nuic et al., 2010). Because accurately modeling flight phases such as climb and descent requires the detailed parameters available in BADA 4, proxy aircraft are used for those types that are not directly modeled.

In pursuit of greater accessibility, OpenAP was explicitly designed for the research community as an open-source model, promoting transparency and broad collaboration (Sun et al., 2020). Recently, Poll and Schumann (2021b,a) proposed a method based on aerodynamic theory and empirical data to predict cruise fuel consumption and performance characteristics for turbofan aircraft.

2.1.2. Machine learning models

Recent studies demonstrated the effectiveness of machine learning over conventional physical models. For instance, Chati and Balakrishnan (2017) used Gaussian process regression and operational data to enhance model accuracy. Building on this, the same authors predicted fuel flow rates using classification and regression trees and least squares boosting, significantly improving emissions inventory accuracy (Chati and Balakrishnan, 2016). Similarly, Baumann and Klingauf (2020) leveraged full-flight sensor data with machine learning methods to outperform traditional fuel flow models. High accuracy in fuel flow prediction was further showcased by Baklacioglu (2021) through the use of advanced neural networks such as radial basis function networks. Evaluating neural network models against specific aircraft types, (Trani et al., 2004) highlighted the potential for real-time simulation applications. Additionally, Li et al. (2021) employed long-short term memory (LSTM) neural networks to accurately predict performance-based contingency fuel.

In the context of exhaust emissions and combustion efficiency, Kayaalp et al. (2021) achieved high accuracy with an LSTM model without extensive experimental testing. Likewise, Metlek (2023) developed a model that surpassed previous methods in accurately predicting aircraft fuel consumption. The integration of genetic algorithm-optimized neural networks by Baklacioglu (2016) introduced a novel approach to fuel consumption prediction across various flight phases. Moreover, Uzun et al. (2021) proposed a hybrid strategy that incorporated physics-based loss during training to enhance model robustness against parameter changes, showing significant promise in making neural network models more resilient. Neural networks were also benchmark for estimating on-board aircraft parameters such as fuel flow and flap configuration during approach and landing (Jarry et al., 2020), which could lead to improved ATM metrics (Jarry and Delahaye, 2021). Additionally, an open-source generic aircraft fuel flow regressor for ADS-B data was released, trained on QAR data (Jarry et al., 2024a).

Despite significant advancements in fuel estimation, as reviewed in the state of the art, most existing models (physics-based or machine learning-driven) struggle to generalize effectively across different aircraft types and operating conditions, especially those not represented in the training data.

2.2. Domain generalization

Accurate estimation of aircraft fuel flow is critical for several applications, including the evaluation of new procedures, the design of next-generation aircraft, and the assessment of the environmental impact of aviation. The challenge is to develop models that can generalize across different aircraft types and operating conditions. To address this challenge, the concept of domain generalization has emerged as a promising approach. This paper is among the first to specifically quantify the generalization capabilities of deep learning models in the context of aircraft fuel flow estimation.

Domain generalization aims to create models that perform well in unseen domains by using training data from multiple source domains (Zhou et al., 2022). This is particularly relevant for aircraft fuel flow estimation, where models must generalize to different aircraft types and flight conditions. Domain alignment techniques are essential for this purpose, as they help minimize the differences between the source domains, allowing the model to learn domain-invariant representations (Muandet et al., 2013; Li et al., 2018a). Techniques such as minimizing moments (mean and variance) (Muandet et al., 2013), contrastive loss (Motiian et al., 2017), Kullback–Leibler (KL) divergence (Li et al., 2020), maximum mean discrepancy (Li et al., 2018a), and domain-adversarial learning (Li et al., 2018b) are instrumental in aligning data distributions and making models more robust to changes in aircraft types and operational scenarios.

Data augmentation plays a critical role in simulating domain shifts to improve model generalization. For aircraft fuel flow estimation, data augmentation can be used to create different training scenarios that mimic different aircraft behaviors, characteristics and environmental conditions. Techniques such as random augmentation networks (Xu et al., 2020) and feature-based augmentation (Zhou et al., 2021) can be particularly effective in expanding the diversity of training data, thereby improving the model's ability to generalize.

Incorporating ensemble learning can further enhance the generalization capabilities of fuel flow models. By combining multiple models with different initializations or training data splits, ensemble methods help to capture a wider range of possible scenarios, thereby improving robustness across different aircraft types. Techniques such as domain-specific neural networks (Ding and Fu, 2017) and domain-specific batch normalization (Liu et al., 2020) are particularly relevant in this context, as they allow better handling of domain-specific characteristics inherent in different aircraft models.

Learning disentangled representations could also be beneficial for aircraft fuel flow estimation, where some features of the model are domain-specific (e.g., aircraft-specific characteristics) while others are domain-agnostic (e.g., general aerodynamic principles). This separation could help the model generalize better across different aircraft types by focusing on the most relevant features for each domain (Khosla et al., 2012; Ilse et al., 2020).

Finally, regularization strategies are critical to improving model robustness by reducing reliance on local features and encouraging the use of global structures. In the context of fuel flow estimation, this could involve regularizing the model to focus on general aerodynamic principles rather than specific data peculiarities of particular aircraft types. Techniques such as iteratively masking over-dominant features (Huang et al., 2020) could be integrated with domain alignment and data augmentation to further improve performance.

While these general domain generalization techniques are valuable, context-specific approaches tailored to aircraft fuel flow estimation could offer even greater benefits. Inductive bias plays a critical role here by embedding domain-specific knowledge into the learning process. For example, physically-informed neural networks (PINNs) (Raissi et al., 2019; Cuomo et al., 2022) can incorporate physical laws and constraints, such as flight dynamics equations (Uzun et al., 2021), directly into the model. This approach not only improves generalization, but also ensures that predictions are consistent with known physical principles, resulting in more reliable and robust fuel flow estimates over a wide range of aircraft and operating conditions.

The issue of robustness and generalization of regression algorithms is also studied in classical statistics when sampling is required to build training data sets. Stratification techniques are used to optimally represent a population by dividing it into distinct subgroups called strata. Equal allocation, which ensures that each stratum has the same sample size, can increase the robustness of estimators by ensuring that under-represented strata are not overlooked (Singh et al., 1996; Barnabas and Sunday, 2014)

Recently, a similar idea has been explored using an advanced reweighting scheme (Steininger et al., 2021). The authors adjust the influence of each data point using kernel density estimation (KDE), giving rare data points more influence on model training. This in turn allows for more robust estimators that are less prone to overfitting on redundant data in the training set.

2.3. Contribution of this paper

This paper aims to demonstrate and quantify the generalization properties of neural networks (Jarry et al., 2020; Jarry and Delahaye, 2021) to accurately predict fuel flow even for unseen aircraft types by applying simple domain generalization techniques such as data augmentation and model regularization.

This paper has contributions in both data processing, data aggregation and modeling :

First, we have extended the aircraft features included in existing models (Jarry et al., 2024a), by integrating more parameters from aircraft characteristics (wing span, reference masses or speeds etc.) and engines (by-pass ratio, rated thrust, reference fuel consumption etc.).

Second, we aggregated a large representative dataset of 101,000 flights with 101 different aircraft types (1000 flights each) divided into two subsets: a primary dataset of 64 aircraft used for training, and a secondary dataset composed of 37 aircraft used for generalization assessment. We applied the BADA 4.2.1 model to obtain an estimate of fuel flow (considered as ground truth).

Third, we defined a pseudo-distance between aircraft types. We designed, applied and analyzed a new uniformization process to resample the data based on their local density. We also proposed and evaluated a dedicated loss for this problem.

Finally, we introduce an inductive bias in our model architecture (with a dedicated last layer to fit our problem) and evaluate the generalization performance of the neural network on the unseen aircraft functions of their distance to the training data. To the best of our knowledge, this is the first paper to demonstrate and quantify the ability of fuel flow models to deal with unseen aircraft types during training and showing better results than baselines like proxy aircraft.

3. Methodology

This section outlines the methodology used to develop a generic model and assess its generalization capabilities for predicting the fuel flow of aircraft types not included in the training set. The process is presented in chronological order: Section 3.1 covers the data collection and preparation; Sections 3.2 and 3.3 introduce the distance metric for evaluating aircraft type similarity and the uniformization process. Section 3.4 describes the neural network model architecture and training process and Section 3.5 presents the baselines compared to the model to evaluate its generalization capabilities.

3.1. Data collection and preparation

To train and evaluate a machine learning model for fuel flow, a labeled dataset (i.e., with both inputs and outputs) of flight states (such as true airspeed, altitude, aircraft mass, etc.) and the corresponding fuel flow is required. Additionally, if a generic model applicable to various aircraft types is desired, the model must be conditioned on aircraft and engine-specific characteristics. These two types of data are described below.

3.1.1. Flight data

ADS-B data were obtained from the OpenSky network (Schäfer et al., 2014), with a focus on flights conducted in 2022. The raw data were resampled at 4-s intervals using the *traffic* library (Olive, 2019), and all flight trajectories were enriched with weather data – specifically, wind and temperature – using the *fastmeteo* library (Sun and Roosenbrand, 2023). By utilizing ADS-B trajectory data, we capture realistic flight variations and operational conditions without the need for extensive data engineering to define BADA input ranges. To enhance the dataset, the derivatives of ground speed and true airspeed were calculated after applying an 8-s moving average smoothing.

It should be noted that the majority of the flights originate from Europe and the United States, regions with robust ground receiver coverage and that is covers 503 airports (origin or destination). For short- to medium-range flights, most selected trajectories include complete flight paths from takeoff to landing. However, for long-range flights, such as those operated by the Airbus A380, some segments may be missing in oceanic regions where receiver coverage is unavailable.

The features extracted to represent the flight state in the model include altitude (ft), vertical rate (ft/min), ground speed (kt), true airspeed (kt), ground acceleration (kt/s), air acceleration (kt/s), and air temperature (K). In addition, the aircraft type has been added to match the corresponding aircraft and engine features described in Section 3.1.2 below.

From this global dataset, 64 BADA aircraft types were selected to form the primary dataset, which is used for training the model, and 37 others to build the secondary dataset used to assess generalization performance. The selection process was conducted manually to balance various criteria, including maintaining family integrity where feasible, ensuring representation of both closely related and distant aircraft types, and including boundary cases such as the A380 to evaluate the model's performance across diverse characteristics. This approach ensures the secondary dataset includes a mix of aircraft types to thoroughly evaluate the model across varying degrees of similarity.

Since only ICAO type is available in the ADS-B data, we apply the full BADA 4.2.1 aircraft type model to randomly selected ADS-B trajectories of its corresponding ICAO type. Point outside the flight envelop are removed by using the BADA4 implementation insuring realistic behaviors.

Each BADA aircraft type accounts for exactly 1000 flights randomly selected. The number of 1000 trajectories was selected based on a 4-fold analysis as an optimal trade-off: while increasing the number of trajectories marginally enhances precision, it also significantly extends data needs and potentially training times. To prevent data leakage, ADS-B flights in the secondary dataset were excluded from the primary dataset. This ensures the model is tested on completely unseen aircraft types, avoiding overlap

Table 1
Aircraft lists for primary and secondary datasets.

Primary Dataset
A300B4-601, A330-321, B752WRR40, EMB-135LR, A300B4-622, A330-341, B753RR, EMB-145ER, A318-112, A350-941, B762ERPW56, EMB-145LR, A319-114, ATR42-500, B762GE50, EMB-145XR, A319-131, ATR72-200, B763ERGE61, EMB-170AR, A320-212, ATR72-500, B763PW60, EMB-170LR, A320-214, ATR72-600, B764ER, EMB-170STD, A320-231, B712HGW21, B772LR, EMB-175AR, A320-232, B73320, B772RR92, EMB-175LR, A320-271N, B73423, B773ERGE115B, EMB-175STD, A321-111, B737W24, B788RR67, EMB-190AR, A321-131, B738W26, B788RR70, EMB-190LR, A330-203, B739ERW26, B789GE75, EMB-190STD, A330-223, B744ERGE, B789RR64, EMB-195AR, A330-243, B744GE, B789RR74, EMB-195LR, A330-301, B748F, EMB-135ER, EMB-195STD
Secondary Dataset
A300B4-203, A340-642, B742RR, F100-650, A300B4-608ST, A380-841, B743PW, F70-620, A310-204, A380-861, B773RR92, MD808120, A310-222, ATR42-300, B788GE67, MD808221, A310-308, ATR42-320, B788GE70, MD808321, A310-322, ATR42-400, B788RR53, MD808720, A310-324, ATR72-210, B788RR64, MD808821, A340-213, B73215, EMB-135BJ-L600, A340-313, B73518, EMB-135BJ-L650, A340-541, B73622, F100-620

Table 2
Aircraft, ADS-B and engine features used as input of the model. Aircraft characteristics were obtained from open data sources, while engine features were sourced from ICAO, FOCA, and FOA emissions databases.

Type	Feature	Unit	Type	Feature	Unit	Type	Feature	Unit
ADS-B & weather	Altitude	ft	Aircraft	Wing Area	m^2	Engine	Engine type	–
	Vertical rate	ft/min		Mass	kg		Number of engines	–
	Ground speed	kt		Span	m		Rated power	hp
	True air speed	kt		Length	m		Rated thrust	kN
	Ground acceleration	kt/s		Maximum Take-Off Weight (MTOW)	kg		Bypass ratio	–
	Air acceleration	kt/s		Zero Empty Weight (OEW)	kg		Pressure ratio	–
	Temperature	K		Maximum Mach Operating (MMO)	mach		Take-off Fuel Flow	kg/s
				Velocity Maximum Operating (VMO)	kt		Climb fuel flow	kg/s
				Altitude Maximum Operating (HMO)	ft		Approach fuel flow	kg/s
							Idle fuel flow	kg/s

that could compromise evaluation integrity. Some ICAO types span both primary and secondary datasets (e.g., A300), while others (e.g., A330, A380) contribute to only one dataset to test a wider range of generalization capabilities. The aircraft types included in each dataset are provided in Table 1.

For each observation in both the primary and secondary datasets, we calculated the estimated fuel flow for six different takeoff masses, ranging from 70% to 95% of the maximum takeoff weight (MTOW), in 5% increments, using the BADA 4.2.1 model (via the *pyBADA* library) (Dalmau Codina et al., 2018). Maximum thrust is assumed during the climb phase without any thrust derate and idle thrust is used during descent.

Finally, both datasets (primary and secondary) were randomly split into training, validation, and testing subsets using an (80%, 10%, 10%) distribution on a flight basis, ensuring the independence of each subset. The train and validation subsets of the primary dataset are used to train and select the model, and the test subset is used to assess the performance on known aircraft. The train and validation subsets of the secondary dataset are reserved to build a final "production" model, which involves retraining the model on all available data to leverage the maximum information for enhanced performance. Only the test subset of the secondary dataset is used to assess the generalization performance on unseen aircraft. This separation clearly distinguishes the quantification of generalization (the objective of the paper) from the creation of a final "production" model with maximized aircraft coverage during training.

3.1.2. Aircraft and engine characteristics

To create a model that generalizes across a wide range of aircraft types, aircraft and engine characteristics were matched to each observation according to the corresponding aircraft type. This includes take-off, climb, approach, and idle fuel flow values from the ICAO Engine Emissions Databank, which remain fixed for each trajectory and serve as contextual inputs to the model. In addition, the TO fuel flow is also used in the final layer of the model to add some inductive bias to ensure that the values remain within a realistic range. Table 2 provides details on the features representing the aircraft and engine characteristics included in each dataset observation.

It is important to note that not all characteristics are available for every aircraft type. The missing values arise due to the differing availability of parameters across engine types—for example, Rated Power is available for turboprop aircraft, while Rated Thrust is provided for jet aircraft. When specific data is missing, it is imputed using linear regression iterative imputation from *scikit-learn* (Pedregosa et al., 2011). This design choice ensures that the model retains critical information related to maximum thrust and other operational parameters essential for accurate fuel flow prediction.

3.2. Distance metric for aircraft type similarity

In order to create a similarity measure between aircraft types, we applied a uniform `QuantileTransformer` from *scikit-learn* on both aircraft and engine characteristics to deal with features with different ranges of values. This method transforms the features

Table 3
Examples of computed distances between a subset of aircraft types.

Aircraft type	A318-112	A319-114	A320-214	A330-243	A340-213	A350-941	A380-841	ATR42-400	ATR72-500
A318-112	0.0								
A319-114	0.11	0.0							
A320-214	0.45	0.39	0.0						
A330-243	1.82	1.78	1.53	0.0					
A340-213	1.68	1.64	1.44	0.88	0.0				
A350-941	1.98	1.93	1.66	0.73	0.91	0.0			
A380-841	2.29	2.24	2.00	0.97	0.79	0.66	0.0		
ATR42-400	1.69	1.72	1.95	3.13	2.93	3.32	3.60	0.0	
ATR72-500	1.58	1.60	1.82	2.97	2.77	3.16	3.44	0.21	0.0

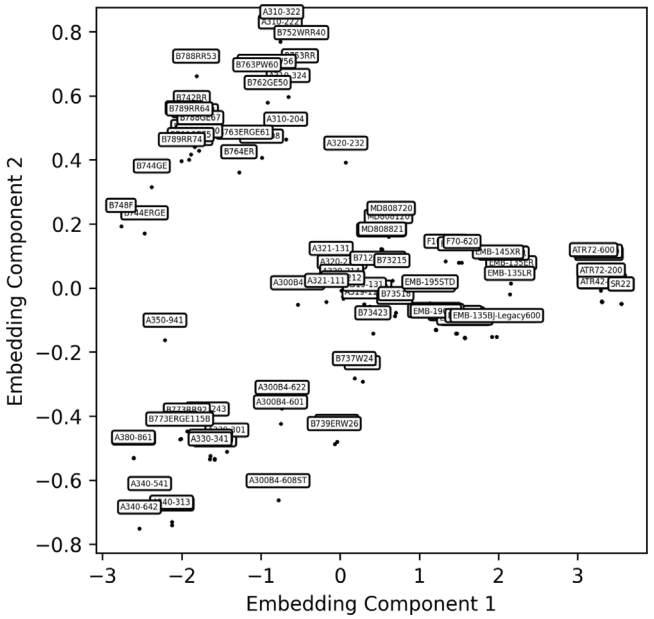


Fig. 1. Illustration of the BADA 4.2.1 aircraft and engines embedded with the `Isomap` process.

to follow a uniform distribution, also reducing the impact of outliers and therefore acting as a robust preprocessing scheme. The similarity measurement is defined as the ℓ_2 norm in the uniform space. Examples of distance are displayed in Table 3

Fig. 1 illustrates the aircraft embedding in two dimensions using an `Isomap` with 10 neighbors and 2 components, applied for illustrative purposes. The `Isomap` method (Tenenbaum et al., 2000) reduces dimensionality by preserving geodesic distances between data points, effectively capturing the intrinsic geometry of the data manifold.

3.3. Uniformization process

To analyze the generalization properties of the model, we compare two types of sampling (at the plot level) on both the training and validation sets, while keeping the test set unchanged. It is worth noting that the training, validation, and test sets were split at the flight level to ensure independence across datasets; specifically, all observations from a given flight are contained within a single set (i.e., either the training, validation, or test set). Applying point-level sampling ensures that models are trained on the same amount of data, which is not guaranteed when splitting on a trajectory basis. This section explains the two subsampling methods used in the paper.

First, random sampling with replacement is proposed: 1000 (respectively 500) x number of trajectories, points are randomly sampled from the entire training (resp. validation) set.

Second, we propose a uniform sampling method applied to the ADS-B parameters for each aircraft type. This process begins by applying a uniform `QuantileTransformer` to the data, followed by dimensionality reduction using Principal Component Analysis (PCA) with 2 components, which projects each observation into a two-dimensional space. Kernel Density Estimation (KDE) is then used to estimate the density of observations in this latent space. The weights for sampling calculated are the inverse of their estimated density, thus favoring observations in less dense regions. The result of this process is shown in Fig. 2. Finally, weighted sampling is performed based on these weights to select a uniform subset of the data according to the specified budget: 1000 (resp. 500) x number of trajectory points for training (resp. validation) set.

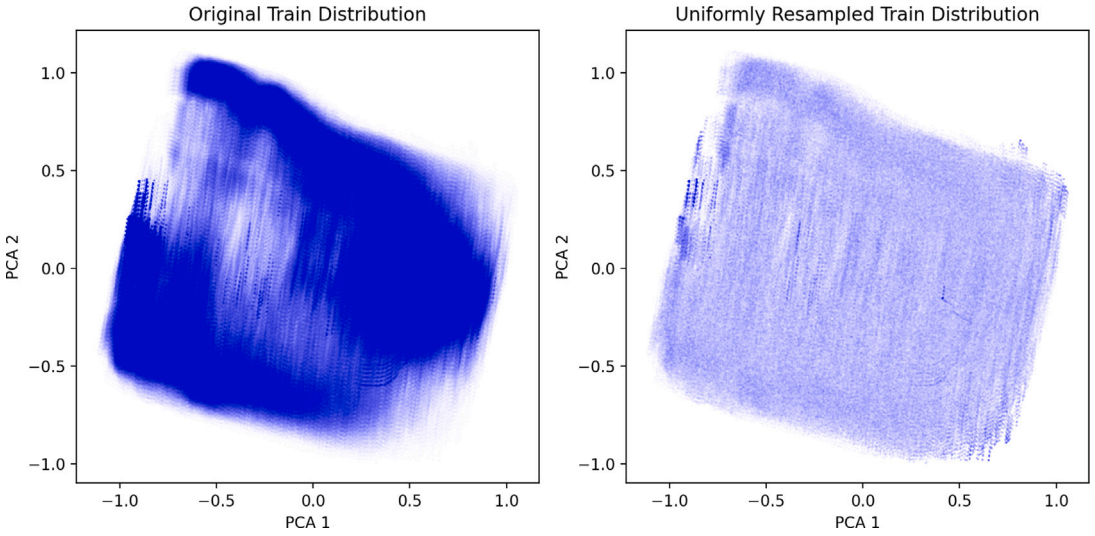


Fig. 2. Example of sample train set distribution in PCA latent space. On the left is the original train distribution, and on the right is the distribution after the uniformization process.

In the end, each sampling strategy provides the same number of observations, but these observations do not follow similar distributions.

3.4. Model architecture and training process

The proposed neural network architecture is an *n-k-m* dense neural network designed for robust and efficient performance in predicting fuel flow. The model starts with a batch normalization layer to standardize the inputs, ensuring stability during training. This is followed by a block of N fully connected layers, each comprising K neurons, using ReLU activation functions to introduce non-linearity. These layers are regularized with an ℓ_2 kernel regularization coefficient of 1×10^{-4}, a value selected through hyperparameter analysis to balance overfitting prevention and model flexibility by penalizing large weights.

After this block, a single dense layer with M neurons is introduced, also using ReLU activation and ℓ_2 regularization for continuity in the network structure.

The final layer is designed to constrain fuel flow predictions to an acceptable range. Three variations are proposed to be tested:

- **Variation (C)**: A ReLU activation followed by a minimum layer that limits the output to 1.1 times the fuel flow at takeoff. This variation ensures that the prediction never exceeds a predefined threshold, helping to bound predictions within a realistic upper limit.
- **Variation (R)**: Similar to variation (C), but the minimum is applied with one before the multiplication by 1.1 times the fuel flow at takeoff that scales the fuel flow.
- **Variation (S)**: A sigmoid activation followed by a multiplication layer that scales the output by 1.1 times the takeoff fuel flow. The sigmoid function is used here to naturally constrain the output between 0 and 1, ensuring that predictions stay within a bounded range, and the scaling ensures the output is adjusted to the expected magnitude. The behavior of the Sigmoïd is slightly different from the ReLU, so it will be interesting to analyze the differences.

These variations in the final layer all ensure that the fuel flow predictions remain within physically plausible limits, preventing the model from outputting unrealistically high or low values. The architecture of the model is shown in Fig. 3.

We trained all models on GPUs (NVIDIA RTX A4500 or RTX A6000 Ada Gen) using Tensorflow for 1000 epochs with a checkpoint on the validation set and evaluated the performance of the models on the test set. To ensure reproducibility, we set the same seed in all experiments (ramdom and np.random seed: 42, tf deterministic ops: 1 and python hash seed: 0).

Different metrics are used to train or evaluate the performance of the model: the Mean Square Error (MSE), the Mean Error (ME), the Mean Absolute Error (MAE) and the Mean Absolute Percentage Error (MAPE).

Let D be a set of input–output pairs (x, y) and h be a model to evaluate, the first three metrics are calculated as follows:

$$\mathbf{MSE}(h, D) = \frac{1}{|D|} \sum_{(x,y)\in D} (h(x) - y)^2 \tag{1}$$

$$\mathbf{ME}(h, D) = \frac{1}{|D|} \sum_{(x,y)\in D} h(x) - y \tag{2}$$

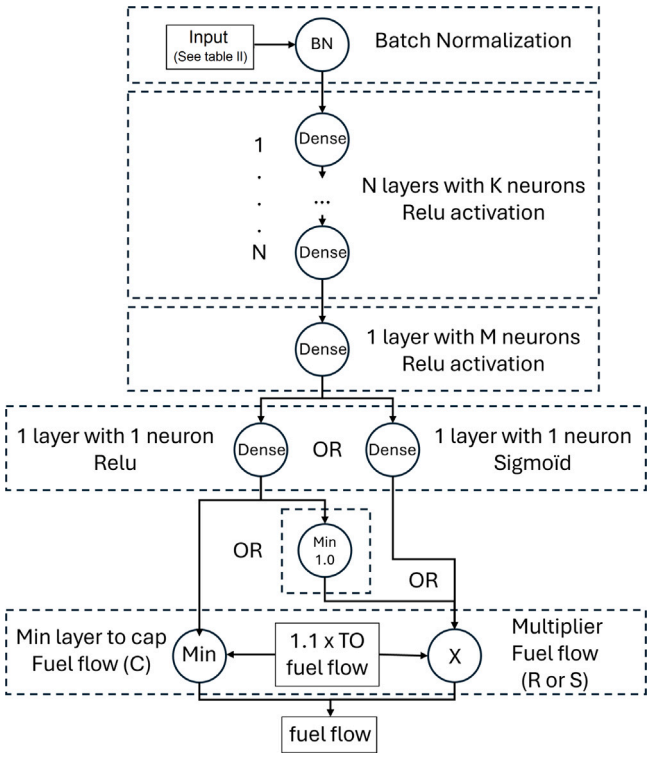


Fig. 3. Illustration of N-K-M dense architecture for fuel flow prediction. The final layer has three variations. Variation (C) consists of a Relu activation and a min layer with 1.1 times fuel flow take-off. Variation (R) consists of a Relu activation, a minimum layer with 1.0 and finally a multiplication layer with 1.1 fuel flow take-off. Finally, variation (S) uses a sigmoid layer and a multiplication layer with 1.1 fuel flow take-off. Input features are described in Table 2.

$$\mathbf{MAE}(h, D) = \frac{1}{|D|} \sum_{(x,y)\in D} |h(x) - y| \tag{3}$$

$$\mathbf{MAPE}(h, D) = \frac{1}{|D|} \sum_{(x,y)\in D} \frac{|h(x) - y|}{y} \tag{4}$$

For MSE, MAE and MAPE, the smaller the value, the more accurate the prediction. For ME, the closer to zero, the better. We also defined a combined loss between MAE and MAPE as :

$$\boldsymbol{\beta}\mathbf{-MAPE}(h, D) = \mathbf{MAE}(h, D) + \beta \times \mathbf{MAPE}(h, D) \tag{5}$$

We designed this loss function to account for the varying range of fuel flow values across different aircraft. By combining MAE, which prioritizes minimizing absolute differences, with MAPE, which focuses on relative percentage errors, we aim to strike a balance between reducing small errors in absolute terms and ensuring that percentage-based deviations are also kept in check. Adjusting the parameter β allows us to fine-tune this balance to suit the characteristics of our data and achieve a compromise between absolute and percentage-based accuracy.

3.5. Baselines

In this study, we will compare the generalization properties of the deep learning model for aircraft fuel flow estimation against two baseline methods. The first baseline is the "proxy aircraft" approach, which estimates the fuel flow of an unseen aircraft by using the closest known aircraft, determined by the pseudo-distance metric defined in Section 3.2, which measures similarity based on physical characteristics and engine parameters. This estimation relies on the BADA model. The second baseline, called the "corrected proxy aircraft", refines this approach by adjusting the fuel flow estimated by BADA with a correction factor based on the ratio of the maximum takeoff weight (MTOW) and the number of engines between the unseen and proxy aircraft. This correction accounts for differences in aircraft size and engine capacity, providing a more refined estimate. These comparisons will allow us to evaluate our model's generalization performance and its accuracy in predicting fuel flow for unseen aircraft types.

Table 4
Performance of the model for **different training loss** on the test set of the primary dataset. The metrics are averaged over all the aircraft types, and the standard deviation is displayed under parenthesis.

Training loss	MAPE (%)	MAE (kg/h)	ME (kg/h)
MAPE	0.99 (0.26)	19.20 (16.23)	−5.01 (8.04)
MAE	0.98 (0.24)	12.78 (7.32)	0.42 (3.51)
MSE	1.28 (0.45)	14.83 (6.21)	6.36 (4.26)
10-MAPE	0.83 (0.18)	13.53 (10.41)	−2.53 (5.85)
20-MAPE	**0.85 (0.19)**	**13.63 (9.86)**	**−0.74 (4.31)**
30-MAPE	0.88 (0.14)	13.94 (10.00)	−1.25 (3.81)

Table 5
Performance of the model for **different batch sizes** on the test set of the primary dataset. The metrics are averaged over all the aircraft types, and the standard deviation is displayed under parenthesis.

Batch size	MAPE (%)	MAE (kg/h)	ME (kg/h)
2500	3.21 (0.52)	40.21 (23.13)	2.58 (5.45)
5000	2.52 (0.42)	34.47 (22.15)	−1.27 (9.46)
10 000	1.78 (0.24)	23.51 (14.21)	0.18 (4.29)
20 000	1.33 (0.20)	17.82 (10.43)	0.02 (2.73)
40 000	1.02 (0.12)	14.26 (8.98)	2.75 (3.30)
80 000	0.89 (0.15)	13.83 (9.68)	−1.75 (3.69)
160 000	**0.85 (0.19)**	**13.63 (9.86)**	**−0.74 (4.31)**
320 000	1.04 (0.19)	17.18 (12.21)	3.49 (4.01)

4. A multi-aircraft unified model

This section is organized into two parts. The first part describes the hyperparameter search and analysis performed on key features. The second part leverages the ability of the model to deal with multi-aircraft in a unified model, testing its performance across 64 known aircraft.

4.1. Training and hyper parameter tuning on primary dataset

To build the reference model, we first conducted a hyperparameter search to identify a suitable configuration. Rather than performing an exhaustive grid search, we focused on finding a robust architecture within a practical timeframe, as each training run took between 2 and 6 h. The goal was not to find the optimal model but to establish a solid reference architecture. For sake of simplicity we will not detail the overall hyperparameter search but focus on main results and interesting findings around loss choice, batch size and sampling process. The remaining details (Learning rate 5e−4, architecture 7-250-4) are available in Appendix A.

4.1.1. Loss

Considering the goal of achieving the best Mean Absolute Percentage Error (MAPE) with minimal bias, the models trained with the β-MAPE loss function show better MAPE performance and reduced Mean Error. The best value seems to be 20, with the model demonstrating an MAPE score of 0.85% (±0.19), and good Mean Error (ME) of −0.74 kg/h (±4.31), suggesting minimal bias and balanced predictions. Although the Mean Absolute Error (MAE) at 13.63 kg/h (±9.86) is not the lowest among the models, it remains within an acceptable range, making the 20-MAPE loss function the most suitable for achieving the desired balance between accuracy and unbiased predictions (see Table 4).

4.1.2. Batch size

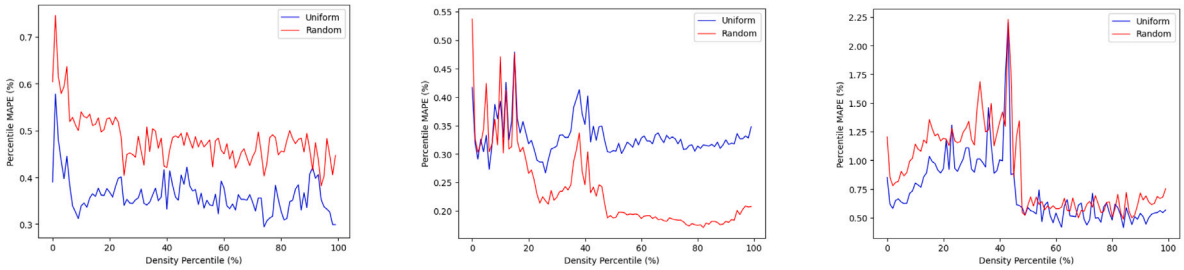
The analysis of model performance across different batch sizes, as presented in Table 5, reveals several key trends in the metrics: Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and Mean Error (ME). As the batch size increases, there is a consistent improvement in MAPE, with values decreasing from 3.21% at a batch size of 2500 to a minimum of 0.851% at 160 000, before slightly increasing again. This indicates that larger batch sizes generally lead to better predictive accuracy. The MAE follows a similar trend, reducing significantly from 40.21 kg/h at 2500 to 13.63 kg/h at 160 000, suggesting improved precision in the model's predictions with larger batch sizes. Notably, this behavior was unexpected.

4.1.3. Sampling process

The analysis of the performance metrics in Table 6 shows that the choice of validation set sampling, whether uniform or random, does not impact the final model selection, as evidenced by identical MAPE, MAE, and ME values for both methods implied same model is selected with the check-pointing. However, the sampling method of the training set seems to influence performance, with uniform sampling yielding slightly better results. Models trained with uniform sampling achieve a lower MAPE of 0.54% than with the random sampling with 0.67% MAPE.

Table 6
Performance of the model for **different train and validation sampling** on the test set of the primary dataset. The metrics are averaged over all the aircraft types, and the standard deviation is displayed under parenthesis.

Train	Validation	MAPE (%)	MAE (kg/h)	ME (kg/h)
Uniform	**Uniform**	**0.54 (0.10)**	**9.08 (7.07)**	**2.55 (3.51)**
Uniform	Random	0.54 (0.10)	9.08 (7.07)	2.55 (3.51)
Random	Uniform	0.67 (0.13)	9.92 (6.52)	0.15 (3.37)
Random	Random	0.67 (0.13)	9.92 (6.52)	0.15 (3.37)



(a) Climb (b) Level (c) Descent

Fig. 4. Performance of the model in MAPE (%) function of density percentile per phases for uniform and random sampling.

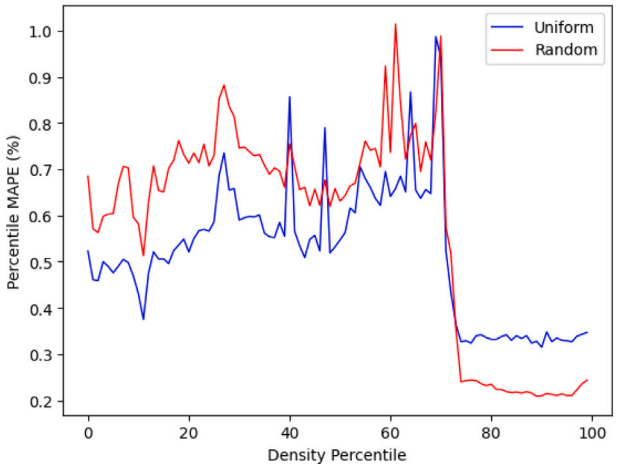


Fig. 5. Performance of the model in MAPE (%) function of density percentile for all phases for uniform and random sampling.

To understand these variations, we analyzed the input distribution per flight attitude (climb, level, descent). In Fig. 6, we show the distribution of the samples as a function of the density percentile in the latent space (cf. Fig. 2) for the different flight attitudes. It can be seen that the high density region is mainly composed of the level attitude, which is consistent with the expected redundancy in this particular phase. The uniformization process then reduces the number of points in level flight to the detriment of climb and descent. If we look at the performance of the two models as a function of density percentile in Fig. 5, we see that the uniformization process increases performance in the low density range and decreases performance in the high density range where performance is actually best. This is consistent with the same graph per flight phase in Fig. 4, where we see that the uniformization process increases the performance of the model for climb and descent and decreases the performance for level flight. If we look at the absolute value, level flight is known to be the easiest phase to predict and indeed shows a low level of MAPE (0.2/0.35%), while descent is the most complex (0.5/2.1%). The uniformization process seems to act as a reweighting of the flight phase and can be beneficial if the aim is to improve performance during ascent and descent.

4.2. Model's ability to manage multi-aircraft in unified architecture

In this section, the model ability to manage multi-aircraft within a single unified architectures is analyzed. On the primary dataset used to train our model (i.e., known aircraft; please refer to Table 1), the average test set error (MAPE) of the best model from the hyper parameter search is approximately 0.54%, with a very small standard deviation of 0.10 across 64 aircraft types. The detailed errors for the individual 64 aircraft are available in Table 7.

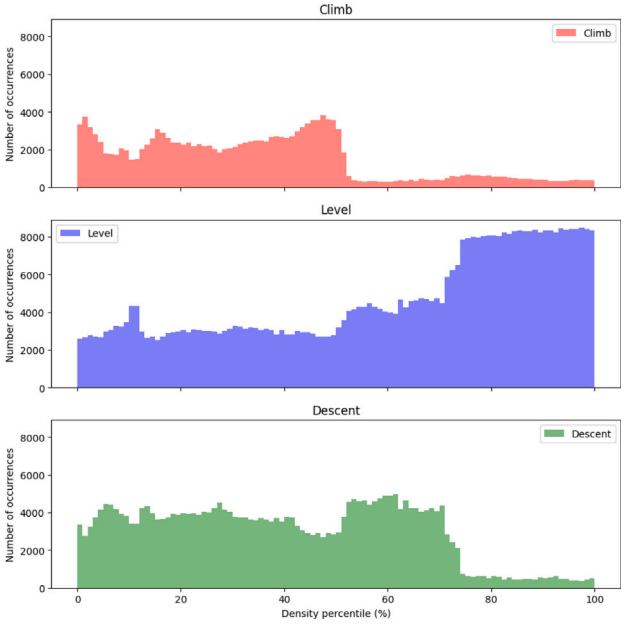


Fig. 6. Histogram of densities per phases of flights.

Table 7
Fuel flow test performance of the best model of the hyperparameter search for the aircraft in primary dataset. The model's performance is evaluated using MAPE, MAE, and ME on the test set, with the standard deviation provided in parentheses. The overall mean and standard deviation for the entire dataset are summarized at the bottom of the table.

Aircraft	MAPE (%)	MAE (kg/h)	ME (kg/h)	Aircraft	MAPE (%)	MAE (kg/h)	ME (kg/h)
A300B4-601	0.52 (1.51)	12.03 (31.40)	4.10 (33.38)	A300B4-622	0.45 (4.20)	10.42 (53.25)	2.92 (54.19)
A318-112	0.59 (1.64)	6.37 (25.39)	−0.83 (26.17)	A319-114	0.61 (2.10)	5.74 (16.43)	0.68 (17.39)
A319-131	0.51 (1.47)	5.35 (13.04)	1.11 (14.05)	A320-212	0.49 (1.53)	4.89 (12.17)	0.80 (13.09)
A320-214	0.51 (1.30)	5.41 (14.90)	0.93 (15.82)	A320-231	0.52 (1.46)	5.11 (14.37)	0.59 (15.24)
A320-232	0.54 (1.78)	5.80 (15.36)	1.66 (16.33)	A320-271N	0.60 (4.24)	5.29 (33.17)	0.49 (33.59)
A321-111	0.48 (1.22)	6.26 (14.18)	2.03 (15.36)	A321-131	0.50 (0.94)	6.27 (14.65)	3.05 (15.64)
A330-203	0.52 (1.62)	12.86 (40.49)	0.16 (42.49)	A330-223	0.45 (1.30)	10.97 (35.88)	2.33 (37.44)
A330-243	0.51 (1.39)	13.26 (41.56)	1.33 (43.60)	A330-301	0.46 (1.28)	10.46 (28.13)	−0.49 (30.00)
A330-321	0.42 (1.26)	9.37 (25.87)	0.92 (27.50)	A330-341	0.50 (1.34)	12.04 (32.60)	1.55 (34.72)
A350-941	0.56 (1.34)	14.63 (32.90)	1.72 (35.97)	ATR42-500	0.50 (0.58)	1.55 (1.70)	0.08 (2.30)
ATR72-200	0.38 (0.70)	1.11 (1.63)	0.43 (1.93)	ATR72-500	0.45 (0.91)	1.29 (2.52)	0.01 (2.84)
ATR72-600	0.50 (0.96)	1.60 (2.27)	0.94 (2.61)	B712HGW21	0.57 (1.37)	5.53 (13.09)	1.89 (14.08)
B73320	0.43 (1.19)	4.40 (10.78)	1.16 (11.59)	B73423	0.55 (1.70)	6.09 (17.53)	1.75 (18.48)
B737W24	0.63 (1.93)	6.00 (19.04)	1.79 (19.88)	B738W26	0.52 (1.37)	5.66 (13.93)	1.57 (14.95)
B739ERW26	0.58 (1.60)	6.71 (19.61)	2.43 (20.59)	B744ERGE	0.74 (2.67)	20.32 (104.27)	1.89 (106.21)
B744GE	0.65 (2.31)	15.47 (57.15)	−0.00 (59.21)	B748F	1.06 (15.18)	22.02 (135.10)	3.31 (136.84)
B752WRR40	0.71 (2.08)	11.44 (28.92)	5.35 (30.64)	B753RR	0.58 (3.08)	10.57 (32.58)	5.43 (33.82)
B762ERPW56	0.65 (2.50)	27.75 (274.39)	19.01 (275.13)	B762GE50	0.48 (1.26)	9.63 (26.69)	2.88 (28.22)
B763ERGE61	0.56 (3.05)	13.96 (41.67)	5.98 (43.54)	B763PW60	0.56 (2.12)	15.70 (159.01)	8.51 (159.55)
B764ER	0.51 (1.55)	13.03 (37.59)	7.30 (39.11)	B772LR	0.72 (2.79)	32.29 (152.14)	13.65 (154.93)
B772RR92	0.53 (1.85)	18.63 (75.53)	3.02 (77.73)	B773ERGE115B	0.68 (3.59)	30.72 (155.62)	10.16 (158.30)
B788RR67	0.62 (1.93)	14.80 (50.91)	5.66 (52.71)	B788RR70	0.64 (2.40)	15.42 (114.49)	5.52 (115.39)
B789GE75	0.61 (2.34)	16.41 (114.60)	6.72 (115.58)	B789RR64	0.56 (1.42)	12.29 (29.12)	3.89 (31.36)
B789RR74	0.70 (2.12)	20.07 (85.88)	9.56 (87.68)	EMB-135ER	0.54 (1.28)	2.53 (4.50)	0.61 (5.12)
EMB-135LR	0.55 (1.45)	2.73 (5.39)	0.23 (6.03)	EMB-145ER	0.54 (1.41)	2.71 (6.11)	−0.25 (6.68)
EMB-145LR	0.53 (1.42)	2.56 (5.05)	0.17 (5.66)	EMB-145XR	0.66 (1.91)	3.50 (6.77)	1.29 (7.51)
EMB-170AR	0.49 (1.91)	3.25 (9.76)	0.14 (10.29)	EMB-170LR	0.45 (1.45)	3.21 (11.91)	−0.06 (12.33)
EMB-170STD	0.46 (1.66)	3.26 (10.25)	−0.36 (10.75)	EMB-175AR	0.45 (1.31)	3.16 (9.44)	0.69 (9.93)
EMB-175LR	0.49 (1.73)	3.39 (15.48)	0.26 (15.84)	EMB-175STD	0.46 (1.75)	3.23 (12.04)	0.28 (12.46)
EMB-190AR	0.51 (1.62)	5.07 (31.03)	1.87 (31.38)	EMB-190LR	0.47 (1.35)	4.58 (19.80)	0.87 (20.30)
EMB-190STD	0.45 (1.27)	3.92 (18.71)	0.58 (19.11)	EMB-195AR	0.45 (1.43)	3.95 (19.67)	0.50 (20.06)
EMB-195LR	0.45 (1.50)	3.69 (16.90)	0.61 (17.29)	EMB-195STD	0.44 (1.45)	3.61 (21.46)	0.58 (21.75)
Mean	0.54 (1.96)	9.08 (39.03)	2.55 (40.15)				
Standard deviation	0.10 (1.82)	7.12 (49.16)	3.54 (49.41)				

Table 8
Performance of the model for **different train sampling** on the secondary dataset. The metrics are averaged over all the aircraft types, and the standard deviation is displayed under parenthesis.

Train	MAPE (%)	MAE (kg/h)	ME (kg/h)
Random	**12.40 (12.29)**	**197.08 (192.89)**	**22.54 (193.04)**
Uniform	13.46 (10.07)	220.09 (185.49)	88.20 (208.14)

Table 9
Performance of the model for **different last layer variation** on the secondary dataset. The metrics are averaged over all the aircraft types, and the standard deviation is displayed under parenthesis.

Last layer	MAPE (%)	MAE (kg/h)	ME (kg/h)
C	20.39 (17.26)	283.27 (224.26)	−9.52 (239.20)
R	**13.46 (10.07)**	**220.09 (185.49)**	**88.20 (208.14)**
S	18.69 (20.76)	249.99 (246.19)	−95.44 (265.22)

These results clearly indicate that our model effectively replicates the BADA4.2 fuel flow estimates, essentially serving as an almost perfect surrogate model. While further statistical tests could be conducted, the exceptionally low average MAPE provides compelling evidence of the model's ability to capture the variability inherent in a large and diverse fleet of aircraft.

5. Generalization performance results

In this section we will assess the generalization performance of our model compared the proxy baselines. The generalization property of the model is evaluated using the same metrics on the secondary dataset (37 unseen aircraft types). To ensure stability in training and testing, we will only change the architecture of the model here and see how the generalization property evolves. As introduced in Section 3.5, the best model will be confronted with two baselines. First, a simple proxy aircraft by calculating the fuel flow of the closest aircraft in distance using BADA. Second, a corrected proxy aircraft by introducing a correction factor to the obtained fuel flow. This correction factor is the ratio of the MTOW to the number of engines of each aircraft.

The task we are attempting is particularly challenging due to the design of our dataset, which excludes entire families of aircraft from the training set, such as the A310, A340, A380, and the F and MD series. We aim to generalize the model's performance to these aircraft families not seen during training. In a more typical scenario, one would expect the model to have been exposed to at least one aircraft from the same family, making it a matter of predicting the performance of a new version or variant of that aircraft. This lack of representations in the training set significantly increases the complexity of the task, as the model must extrapolate from characteristics and behaviors it has never encountered before. Our study aims to demonstrate and quantify the model's ability to overcome this challenge and provide accurate predictions even under these adverse conditions.

This section is divided into three subsections. The first subsection analyzes the impact of three features on the generalization property (the sampling process, the last layer variation, and the introduction of noise regularization). The second subpart analyses and quantifies the global generalization performance and dives into some aircraft results. Finally, the third subsection showcase a per input parameter analysis to better understand model behavior.

5.1. Per feature analysis

In the section, the generalization property is analyzed while changing some architecture or process features.

5.1.1. Sampling process

As shown in Table 8, the uniform sampling procedure seems to produce a slight decrease effect on the generalization properties of the models, with a MAPE of 13.46% for uniform sampling versus 12.40% for random sampling.

5.1.2. Last layer variation

Regarding the architectures in Table 9, we observe that the Relu (R) architecture presents the best results with 11% MAPE, but if we check the evolution of the generalization performance during the learning process, we observe that it is really chaotic (blue curve, as shown in Fig. 7). In particular, we see that during training, the MAPE on the secondary data set increases and decreases, indicating a possible overfitting with respect to the aircraft and engine characteristics, while the validation curves (on the primary data set) show no sign of overfitting. Since the checkpoint is applied on the validation set we do not have any insurance that the learning stops during bad generalization performance.

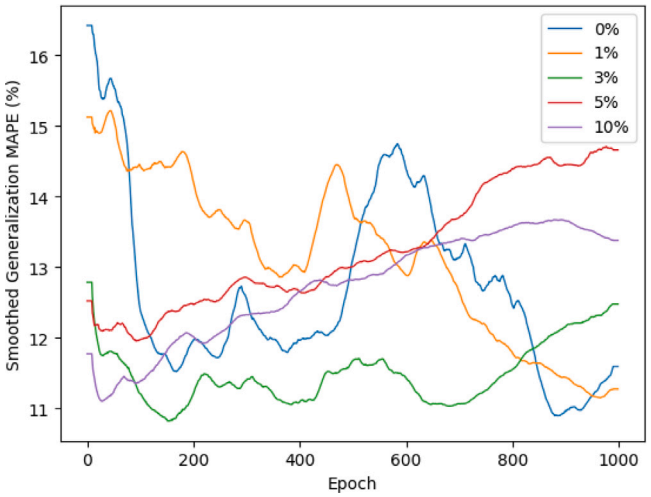


Fig. 7. Generalization MAPE performance (%) curve during training process for different noise levels.

Table 10
Performance of the model **with or without input noise** on the secondary dataset. The metrics are averaged over all the aircraft types, and the standard deviation is displayed under parenthesis.

Noise p	MAPE (%)	MAE (kg/h)	ME (kg/h)
0%	13.46 (10.07)	220.09 (185.49)	88.20 (208.14)
1%	**10.34 (5.57)**	**181.63 (162.72)**	**40.40 (175.82)**
3%	10.96 (6.13)	183.96 (166.75)	59.65 (169.79)
5%	13.00 (9.66)	212.33 (179.87)	71.07 (186.44)
10%	12.15 (10.88)	207.86 (192.69)	45.46 (204.17)

5.1.3. Noise regularization

To try to mitigate this behavior, noise regularization is applied to the input aircraft and engine features. For each feature, we sampled a Gaussian distribution with a mean $\mu = 0$ and a standard deviation $\sigma = 0.33$.

Each noisy feature $\hat{X}_p$ is computed as

$$\hat{X}_p = X \cdot \left(1 + p \cdot \mathcal{N}(0, 0.33^2)\right)$$

where X is the original feature and p is the noise percentage parameter. As observed in Table 10, introducing noise improves model performance, especially at noise levels of 1% and 3%, where both MAPE and MAE show significant reductions. We also observe more stability in the MAPE curve during training and in particular with 1% noise a proper decrease of the MAPE curve without overfitting periods. However, when the noise level is too high, such as at 10%, it essentially overwhelms and distorts the underlying information in the data. This explains why the MAPE curve with a high degree of noise begins to increase with the training epochs, as the model struggles to extract meaningful patterns from the corrupted input, leading to deteriorating performance (see Fig. 7).

5.2. Statistical analysis of generalization performance

In this section, a statistical analysis of global generalization property is performed. First, we observe that the model has a lower average MAPE (9.22% vs. 16.11%) and a lower standard deviation (5.53% vs. 10.73%). This shows a clear overall advantage for the model over proxy approaches. More specifically, the model has a lower error in 70% of the aircraft types as shown in Table 11 with an average of 10% less error than the baselines. Moreover, when the model gives the worst results, it has, on average, only 2.08% more errors than the baselines. This tendency is shown in Fig. 9 and Table 11. Overall, we observe a linear trend in the performance MAPE as a function of the distance to the closest aircraft for our model, whereas the baselines exhibit a much steeper increase with distance. The neural network-based model seems to demonstrates greater robustness and significantly fewer extreme error values, particularly for aircraft types that are farther from the training set in terms of characteristics (see Fig. 8).

To better statistically confirm these results a Wilcoxon signed-rank test is performed. It is a non-parametric method used to compare the performance metrics of our model and the proxy method. It tests whether there is a statistically significant difference in favor of the model. Let us consider the following hypothesis:

- (H_0): There is no difference between the performance of our model and the proxy method (i.e., the median difference is zero).
- (H_1): Our model outperforms the proxy method (i.e. the differences are consistently positive when the proxy performance is subtracted from the model performance).

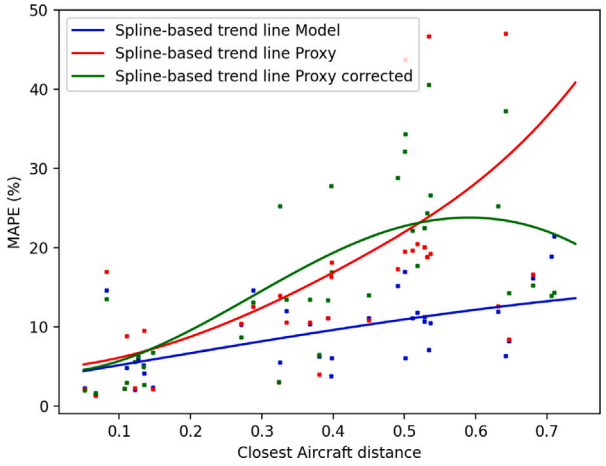


Fig. 8. MAPE performance (%) of the best model (with 1% noise) compared to baselines on secondary dataset function of the closest aircraft in the primary dataset. Each point is one of the 37 aircraft in the secondary dataset. Trend lines are computed using smoothing spline interpolation with a smoothing factor of 20.

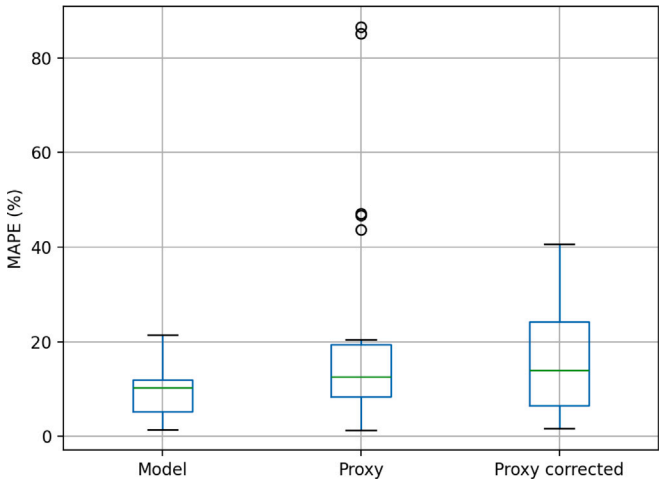


Fig. 9. MAPE performance (%) of the best model (with 1% noise) on secondary dataset function of the closest aircraft in the primary dataset. Each point is one of the 37 aircraft in the secondary dataset.

The test was performed using the MAPE of our model and the proxy method for each point in the generalization test set. The data set consists of 37 991 282 points. The obtained p-value is 0.0 and the z-score is 2941.51684. Since the p-value ($p < 0.01$) is lower than the chosen significance level (1% very conservative), we reject the null hypothesis H_0 in favor of the alternative hypothesis H_1. This indicates that the observed effect is statistically significant and that the probability of obtaining these results by chance is less than 1%.

In addition, a common way to assess the strength of a model is to use the effect size, r. This is a standardized measure that quantifies the size of an effect regardless of the sample size. In the context of non-parametric tests such as the Wilcoxon signed-rank test, r is typically calculated by dividing the z-score by the square root of the total number of observations. This calculation provides a clear metric for understanding the strength of the effect. Substituting our values gives an effect size of approximately 0.477. Given that common benchmarks define an effect size of 0.1 as small, 0.3 as medium and 0.5 as large, our result suggests a medium to large impact of our methodology.

Finally, to quantify wether the use of our methodology presents an improvement for one of a main use cases: global environmental impact assessment. We calculated the total fuel consumption of the generalization set (39,000 flights) using both methods and compared it to the reference value. The proxy method shows an error in the total fuel consumption of 590 tons of fuel (2.832% error on a total consumption of 20 833 tons of fuel), while the model shows an error of 448 tons of fuel (2.1504%). The use of our methodology shows a reduced error in the global assessment of fuel consumption.

Table 11
Fuel flow generalization performance of the best model for the aircraft in secondary dataset compared to the proxy and proxy corrected baselines. Model are compared based on their MAPE on the test set (%). The mean and standard deviation for the entire dataset are displayed at the bottom of the table. Aircraft in orange are those where the proxy corrected model is having lower MAPE than the model.

Aircraft	Closest aircraft	Distance	Model MAPE (%)	Proxy MAPE (%)	Proxy corrected MAPE (%)
A300B4-203	A300B4-622	0.334	11.96	10.53	13.47
A300B4-608ST	A300B4-601	0.531	18.85	18.82	24.33
A310-204	B762GE50	0.380	6.30	3.99	6.46
A310-222	B752WRR40	0.397	6.05	18.11	16.91
A310-308	A300B4-601	0.367	10.35	10.54	13.46
A310-322	B752WRR40	0.397	3.75	16.33	27.83
A310-324	B753RR	0.325	5.54	13.91	25.26
A340-213	A330-321	0.710	21.42	85.10	14.35
A340-313	A330-321	0.706	18.89	86.58	13.92
A340-541	B748F	0.680	16.12	16.61	15.25
A340-642	B748F	0.646	8.23	8.41	14.30
A380-841	B748F	0.490	15.17	17.32	28.82
A380-861	B748F	0.500	16.95	19.50	32.12
ATR42-300	ATR42-500	0.135	4.14	9.54	2.67
ATR42-320	ATR42-500	0.110	4.80	8.80	2.97
ATR42-400	ATR42-500	0.082	14.61	16.93	13.53
ATR72-210	ATR72-500	0.051	2.25	2.10	2.01
B73215	B73320	0.631	11.90	12.65	25.23
B73518	B73320	0.066	1.42	1.33	1.66
B73622	B737W24	0.146	2.37	2.11	6.75
B742RR	B744GE	0.450	11.06	10.84	14.03
B743PW	B744GE	0.392	11.12	11.13	13.37
B773RR92	B772RR92	0.121	2.08	2.23	5.59
B788GE67	B788RR67	0.133	5.18	4.95	4.95
B788GE70	B788RR70	0.126	5.86	6.26	6.26
B788RR53	B788RR67	0.323	3.04	3.06	3.06
B788RR64	B788RR67	0.107	2.20	2.20	2.20
EMB-135BJ-Legacy600	EMB-145XR	0.271	10.30	10.37	8.66
EMB-135BJ-Legacy650	EMB-145XR	0.287	14.59	12.57	13.12
F100-620	EMB-145XR	0.534	7.12	46.68	40.57
F100-650	EMB-145XR	0.642	6.35	47.06	37.28
F70-620	EMB-145XR	0.501	6.07	43.70	34.33
MD808120	A319-131	0.536	10.47	19.23	26.63
MD808221	A319-131	0.527	11.22	20.07	22.48
MD808321	A319-131	0.518	11.82	20.50	17.71
MD808720	A319-131	0.510	11.08	19.68	22.12
MD808821	A319-131	0.527	10.69	20.04	22.46
Mean		0.383	9.22	18.37	16.11
Standard deviation		0.203	5.35	19.83	10.73

5.3. Comparative analysis of different methods results

In this section, we analyze the discrepancies between our proposed method and the proxy method. We provide a per-feature evaluation, and delve into discrepancies observed for certain aircraft types.

For each studied input parameter, we computed a histogram on the primary dataset to determine the distribution of values and identify the appropriate bin edges. These bins were then used to segment the data into five groups based on specific value intervals for the parameter. The same bin boundaries were subsequently applied to the secondary dataset to extract and visualize the corresponding error metrics using boxplots. This approach enables a direct comparison of the availability of training samples with the performance errors across different value ranges of each parameter.

For the sake of readability, this section focuses on three parameters: maximum take-off weight, maximum operating altitude, and number of engines. Additional graphs are available in Appendix B. Overall, Figs. 10 to 12 align with our previous analysis, consistently showing that the proposed method achieves lower error rates and reduced standard deviations compared to the proxy-corrected approach.

This subgroup analysis also highlights the intrinsic noise and challenges in predicting fuel flow for unknown aircraft types, even when using a proxy-corrected approach. These findings underscore that the relationship between fuel flow and aircraft parameters is non-linear and is more effectively captured by our neural network model than by a simple proxy approach with correction coefficients. Notably, the subgroup characterized by a maximum take-off weight of 230 tonnes and the subgroup of four-engine aircraft reveal that our model exhibits a higher MAPE than the proxy approach for the A340 family. Similarly, the subgroup corresponding to a maximum operating altitude of 27,010 ft indicates a similar trend for the ATR42 family.

In the following, we analyze aircraft for which our model exhibits higher MAPE than the proxy-corrected method. These aircraft are highlighted in orange in Table 11 and can be grouped into three categories:

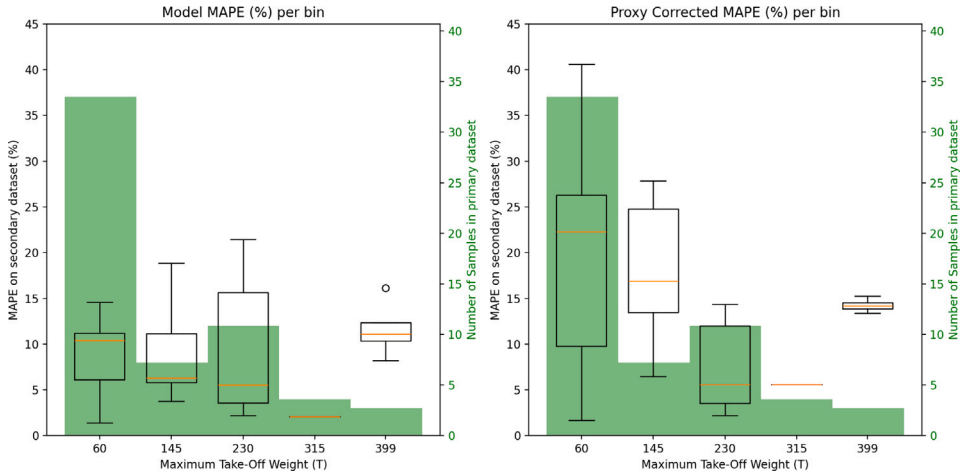


Fig. 10. Maximum take-off weight groups: MAPE boxplots for the Model (left) and the Proxy-corrected method (right), along with the corresponding training sample counts. The groups were defined based on histogram binning of the primary dataset, and the same bin edges were applied to the secondary dataset to enable a direct comparison of performance across bin intervals.

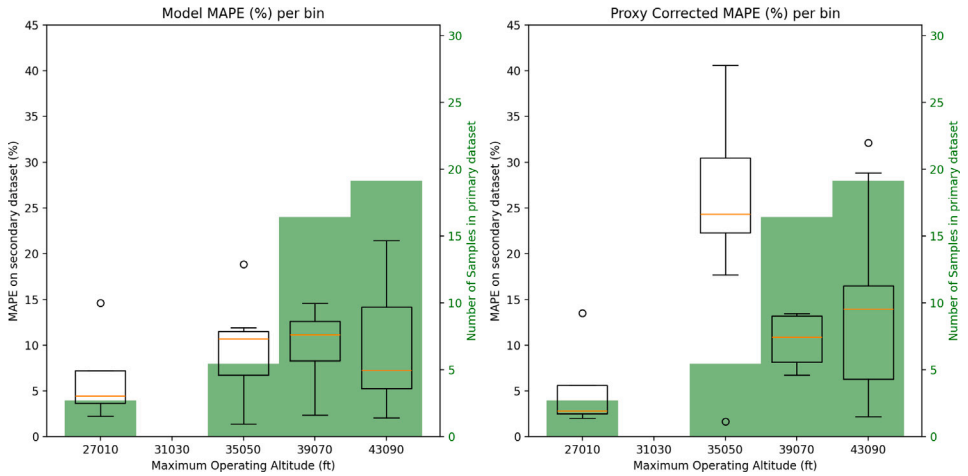


Fig. 11. Maximum operating altitude groups: MAPE boxplots for the Model (left) and the Proxy-corrected method (right), along with the corresponding training sample counts. The groups were defined based on histogram binning of the primary dataset, and the same bin edges were applied to the secondary dataset to enable a direct comparison of performance across bin intervals.

(1) **Minimal Difference (Noise Level):** For some aircraft, the difference in MAPE is very small (less than 1%), which we attribute primarily to noise. Examples in this category include B788GE67, ATR72-210, and A340-541.

(2) **A340 and ATR42 Families:** These aircraft are particularly interesting because, while the raw proxy model initially performed worse than our method, applying corrections based on the number of engines and maximum take-off weight allowed the proxy approach to outperform our model. For the A340 family, this appears to be mainly due to the proxy model using a lower number of engines than appropriate. In the case of the ATR42 family, the weight correction was the critical factor. It is important to note that our training dataset includes only one turboprop aircraft (the ATR72-500) and only the B747 family with four engines, which likely contributes to the model's difficulties with these families. This suggests that the distribution of aircraft in the primary dataset versus the secondary dataset plays a significant role.

(3) **EMB-135BJ-Legacy Family:** The performance differences for this family are more complex. On one hand, the MTOW correction enhanced the proxy performance for the Legacy600, whereas it decreased the performance for the Legacy650. Additionally, similar aircraft were present in the primary dataset, further complicating the analysis. This indicates that the underlying data distribution and representation of these aircraft types might be influencing the results in non-trivial ways.

These analyses highlight the challenges in modeling fuel flow for certain aircraft families, particularly when the available training data is not evenly distributed across all types.

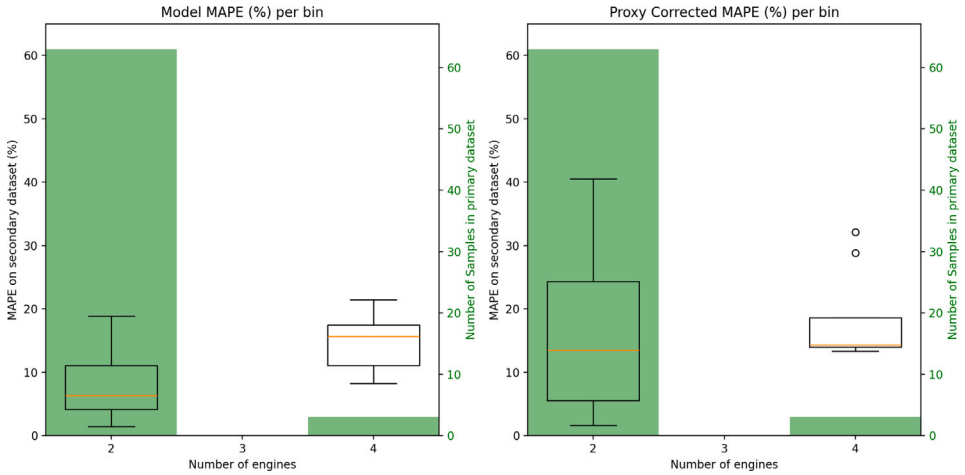


Fig. 12. Number of engine groups: MAPE boxplots for the Model (left) and the Proxy-corrected method (right), along with the corresponding training sample counts. The groups were defined based on histogram binning of the primary dataset, and the same bin edges were applied to the secondary dataset to enable a direct comparison of performance across bin intervals.

6. Discussion

A limitation of our current approach is the reliance on the BADA physical model as a proxy for estimating fuel flow. In reality, our goal is to develop and release models trained on Quick Access Recorder (QAR) data. However, the availability of these data is not sufficient and representative enough to conduct such an analysis at scale. The BADA model is known to provide consistent estimates of the fuel flow and physically consistent outputs. It can, therefore, be assumed that the global correlation between estimated fuel flow from BADA and aircraft and engine parameters will be respected.

The fuel data estimated using BADA appears to be significantly less noisy, which contributes to the exceptionally good performance (with errors less than 1%) observed in our test results on the primary dataset. Indeed, the model build is a surrogate model of an existing physical based model. However, the aim of this study is not attesting the accuracy of the model but its capability to deal with unseen aircraft types. The results obtained confirmed that this type of model is more robust when predicting fuel flow on unseen aircraft types. Finally, the transition to models trained on QAR data will provide more accurate and reliable fuel flow estimates that better reflect real-world conditions.

In this study, we used ADS-B trajectory data and applied the BADA model to it to generate a training dataset. This integration captures realistic flight conditions and operational variations without the extensive data engineering required to manually define BADA input ranges. However, our model currently focuses on nominally observed flight envelopes, meaning that it does not consider every point within the full flight envelope during training, such as extreme or atypical conditions. To address this limitation, in future work we will integrate physically informed loss functions to ensure that the model responds appropriately to all possible conditions.

Another limitation of our current approach is the discrete sampling of mass values, as mentioned in the paper. This method could be improved by implementing a random sampling technique for mass values, which would provide a more continuous representation. In addition, a more appropriate procedure would be to use historical mass distributions specific to each city pair. This improvement would allow for more accurate and realistic modeling of aircraft fuel flow that more closely reflects actual operational conditions. In addition, the current process is limited to 1000 trajectories. Improving this by maximizing the variability within those 1000 trajectories could result in a more robust and comprehensive model that captures a wider range of flight conditions and operational scenarios.

An interesting aspect of our results relates to the overfitting observed on the aircraft and engine parameters. The model does not appear to overfit on the ADS-B parameters, where it is exposed to millions of data points, indicating robust generalization capabilities in this context. However, there is a noticeable tendency to overfit on the aircraft and engine parameters, which are limited to only 64 possible values. This discrepancy suggests that the model struggles with the limited variability of these features, leading to overfitting. Interestingly, introducing noise into the aircraft and engine parameters effectively alleviated this problem. By adding noise, we increase variability and prevent the model from learning spurious patterns specific to the training set, thereby improving its generalization performance to unseen aircraft types.

Furthermore, We deliberately chose a modern paradigm with a single neural network to allow the model to benefit from shared knowledge across all aircraft types, leveraging common operational principles to enhance overall performance. The aim of this paper is not to prove that a unified approach is superior but to demonstrate its ability to generalize effectively across diverse aircraft types, providing a robust framework for fuel flow prediction.

Future work could explore improvements in the input features to better capture the evolution of engine technologies. For instance, incorporating the year of Entry-into-Service as a feature or the Thrust Specific Fuel Consumption (TSFC) could help

Table 12
Table showing the performance of the model for **different learning rates** on the test set of the primary dataset. The metrics are averaged over all the aircraft types, and standard deviations are displayed under parenthesis.

Learning rate	MAPE (%)	MAE (kg/h)	ME (kg/h)
1e−04	1.44 (0.37)	22.73 (14.82)	6.04 (7.43)
5e−04	**0.85 (0.19)**	**13.63 (9.86)**	**−0.74 (4.31)**
1e−05	2.60 (0.59)	42.04 (29.32)	13.95 (14.22)

improve the model's ability to extrapolate across aircraft generations. Improvements in the model architecture could also introduce more inductive bias and enhance generalization. For example, training an embedding through adversarial learning could help the model learn more robust representations. Another architectural enhancement could involve explicitly modeling TSFC, leveraging its variation under different flight conditions to better capture underlying engine performance trends. Finally, advancements in the output modeling could focus on learning the fuel flow distribution functions specific to each aircraft. This would provide more context-aware predictions and improve the model's applicability across a wide range of scenarios.

7. Conclusions

In this paper, we explored the generalization properties of neural networks for aircraft fuel flow estimation, focusing on their ability to predict fuel flow for aircraft types not included in the training data.

Our results highlighted the beneficial impact of sampling methods. Uniform sampling of the training data improved model performance during the ascent and descent phases, which are more complex and typically underrepresented in the raw data.

The generalization capabilities of deep learning models for aircraft fuel flow estimation show significant potential for extending their applicability to unseen aircraft types. By integrating domain generalization techniques such as data augmentation, uniform sampling, and noise regularization, the models demonstrated improved robustness and accuracy when trained on diverse aircraft characteristics. While traditional proxy methods experience performance degradation as the distance between known and unseen aircraft types increases, the deep learning models maintained more consistent and reliable results. Additionally, the introduction of noise into aircraft and engine features helped mitigate overfitting.

Looking forward, this model could further benefit from incorporating domain-specific knowledge, such as physical constraints and historical mass distributions, into the training process. Future work will aim to replace BADA-derived parameters and trajectory data with actual Quick Access Recorder (QAR) data, both for flight path and fuel flow measurements. This transition is expected to improve data accuracy and comprehensiveness. Additionally, integrating QAR-derived parameters, such as aircraft mass, into the training process, whenever available, could further enhance model performance. Future work should focus on refining these techniques and transitioning from proxy models like BADA to direct use of Quick Access Recorder (QAR) data, which would provide more accurate and comprehensive datasets.

In this context, it will be important to investigate the implications of fuel flow estimation errors on ATM applications, including establishing thresholds for acceptable error levels. Moreover, comparing the precision of BADA fuel flow estimates with QAR values for training purposes, and exploring how factors such as entry-into-service dates and engine counts correlate with current errors. Finally, we will explore the use of such surrogate model to assess contrail impact uncertainties.

In addition, a complete BADA 4.2.1 (all 103 aircraft types) surrogate model has been built. Discussions are underway to release the model under a BADA license with an implementation in the DeepEnv library (Jarry et al., 2024b)

CRediT authorship contribution statement

Gabriel Jarry: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Project administration, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Ramon Dalmau:** Writing – review & editing, Writing – original draft, Methodology, Formal analysis, Conceptualization. **Philippe Very:** Writing – review & editing, Writing – original draft, Validation, Software, Project administration, Methodology, Investigation, Formal analysis, Conceptualization. **Junzi Sun:** Writing – review & editing, Writing – original draft, Methodology, Data curation, Conceptualization.

Appendix A. Hyperparameter tuning

Hyperparameter tuning results not described in the paper are detailed in this appendix.

A.1. Learning rate

The analysis in Table 12 shows that a learning rate of 5×10^{-4} provides the best performance, with a MAPE of 0.85%, MAE of 13.63 kg/h, and ME of −0.74 kg/h. In comparison, 1×10^{-4} results in a MAPE of 1.44%, MAE of 22.73 kg/h, and ME of 6.04 kg/h. The lowest rate, 1×10^{-5}, performs worst, with a MAPE of 2.60%, MAE of 42.04 kg/h, and ME of 13.95 kg/h. Hence, 5×10^{-4} is optimal for accuracy and bias.

Table 13
Performance of the model for **different values of N layers in the middle layers** on the test set of the primary dataset. The metrics are averaged over all the aircraft types, and the standard deviation is displayed under parenthesis.

N layers	MAPE (%)	MAE (kg/h)	ME (kg/h)
4	0.87 (0.15)	14.19 (10.53)	−2.09 (3.49)
5	0.71 (0.13)	11.15 (8.39)	0.84 (2.96)
6	0.69 (0.11)	10.24 (7.02)	−3.22 (2.81)
7	**0.66 (0.11)**	**9.79 (6.92)**	**−1.45 (3.50)**
8	0.66 (0.14)	9.73 (6.50)	−0.95 (2.46)
9	0.66 (0.13)	9.90 (7.03)	1.42 (3.71)
10	0.63 (0.11)	9.64 (6.80)	3.93 (4.24)

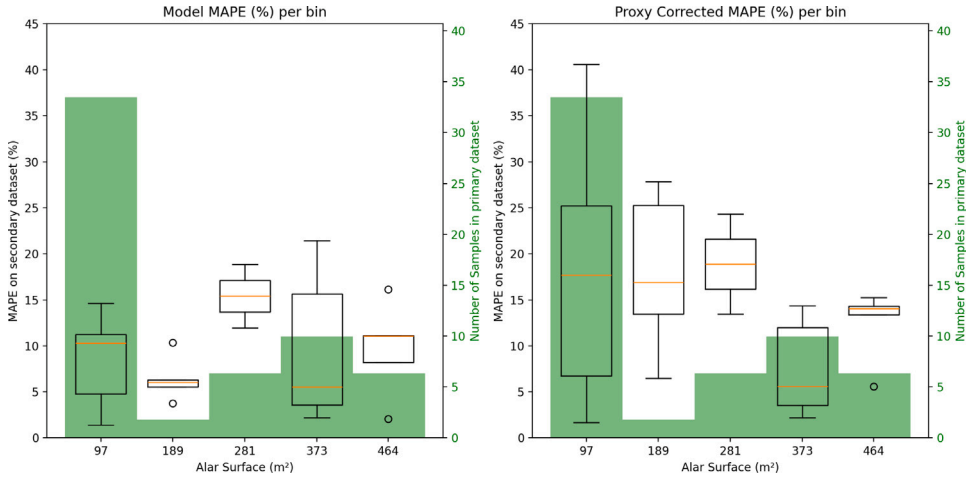


Fig. 13. Alar Surface groups: MAPE boxplots for the Model (left) and the Proxy-corrected method (right), along with the corresponding training sample counts. The groups were defined based on histogram binning of the primary dataset, and the same bin edges were applied to the secondary dataset to enable a direct comparison of performance across bin intervals.

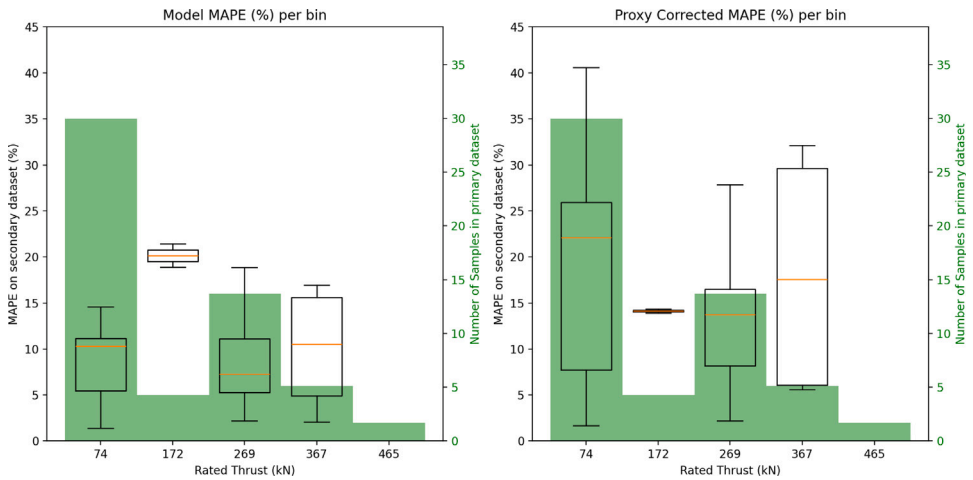


Fig. 14. Rated thrust groups: MAPE boxplots for the Model (left) and the Proxy-corrected method (right), along with the corresponding training sample counts. The groups were defined based on histogram binning of the primary dataset, and the same bin edges were applied to the secondary dataset to enable a direct comparison of performance across bin intervals.

Table A.14
Performance of the model for **different values of K neurons in the middle layers** on the test set of the primary dataset. The metrics are averaged over all the aircraft types, and the standard deviation is displayed under parenthesis.

K neurons	MAPE (%)	MAE (kg/h)	ME (kg/h)
50	1.98 (0.43)	31.18 (19.97)	5.87 (7.79)
100	1.14 (0.23)	18.11 (13.23)	1.39 (3.34)
150	0.85 (0.19)	13.63 (9.86)	−0.74 (4.31)
200	0.66 (0.15)	10.38 (7.40)	2.80 (3.39)
250	**0.57 (0.10)**	**9.14 (6.72)**	**1.68 (2.84)**
300	0.56 (0.10)	8.77 (6.26)	4.15 (4.57)

Table A.15
Performance of the model for **different values of M neurons in the last layer of the middle block** on the test set of the primary dataset. The metrics are averaged over all the aircraft types, and the standard deviation is displayed under parenthesis.

M neurons	MAPE (%)	MAE (kg/h)	ME (kg/h)
2	0.57 (0.10)	9.14 (6.72)	1.68 (2.84)
3	0.58 (0.15)	8.56 (5.88)	−0.21 (2.85)
4	**0.54 (0.10)**	**8.35 (6.06)**	**2.68 (3.79)**
5	0.54 (0.12)	8.99 (7.26)	3.38 (4.86)
10	0.61 (0.11)	9.68 (6.91)	3.55 (3.24)
20	0.57 (0.12)	9.60 (7.95)	4.18 (4.99)
40	0.56 (0.10)	8.71 (6.56)	2.62 (4.70)

A.2. Number of layers

Table 13 compares the performance of models with varying numbers of middle layers, evaluated using Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and Mean Error (ME). The model with 7 layers provides a good balance, showing a low MAPE of 0.66% (±0.11) and a relatively low MAE of 9.73 kg/h (±6.92), with an ME of −1.45 kg/h (±3.50). Although the model with 10 layers has a slightly better MAPE of 0.63% (±0.11) and MAE of 9.64 kg/h (±6.80), it introduces complexity and potential extra training time. Thus, the model with 7 layers is the best compromise between performance and complexity, providing efficient training time while maintaining excellent accuracy.

A.3. Number of neurons K

The performance of the model with varying numbers of neurons in the middle layers, as detailed in Table A.14, highlights significant improvements in predictive accuracy and precision with an increasing number of neurons. The Mean Absolute Percentage Error (MAPE) decreases from 1.98% at 50 neurons to a low of 0.56% at 300 neurons, indicating a clear enhancement in the model's accuracy as the number of neurons grows. Similarly, the Mean Absolute Error (MAE) shows a substantial reduction, dropping from 31.18 kg/h at 50 neurons to 8.77 kg/h at 300 neurons, reflecting increased precision in the model's predictions. The Mean Error (ME), however, presents more variability: it starts at 5.87 kg/h at 50 neurons, decreases to near zero at 150 neurons, and then fluctuates at higher neuron counts, with a notable increase to 4.15 kg/h at 300 neurons. Therefore we selected 250 neurones.

A.4. Number of neurons M

Table A.15 compares the performance of models with varying numbers of neurons (M) in the last layer of the middle block, evaluated using Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE), and Mean Error (ME). The model with 4 neurons exhibits the best performance, with the lowest MAPE at 0.54% (±0.10) and MAE at 8.35 kg/h (±6.06), alongside an ME of 2.68 kg/h (±3.79), indicating minimal bias. Although models with 5 neurons show similar MAPE (0.54% ±0.12), but have higher MAE and ME. Models with more neurons (10, 20, and 40) tend to increase complexity without substantial gains in performance. Therefore, the model with 4 neurons offers the best compromise between performance and model complexity, ensuring efficient training times while maintaining high accuracy.

Appendix B. Per parameter model performance analysis graphs

This appendix contains graphs that were not displayed in Section 5 (see Figs. 13–15).

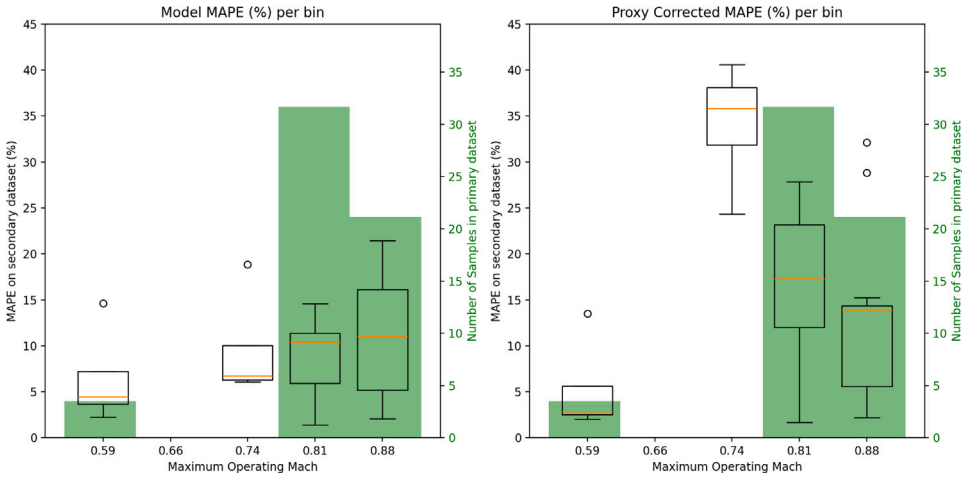


Fig. 15. Maximum operating mach groups: MAPE boxplots for the Model (left) and the Proxy-corrected method (right), along with the corresponding training sample counts. The groups were defined based on histogram binning of the primary dataset, and the same bin edges were applied to the secondary dataset to enable a direct comparison of performance across bin intervals.

References

Baklacioglu, T., 2016. Modeling the fuel flow-rate of transport aircraft during flight phases using genetic algorithm-optimized neural networks. Aerosp. Sci. Technol. 49, 52–62.

Baklacioglu, T., 2021. Predicting the fuel flow rate of commercial aircraft via multilayer perceptron, radial basis function and ANFIS artificial neural networks. Aeronaut. J. 125 (1285), 453–471.

Barnabas, A.F., Sunday, A.O., 2014. Comparison of allocation procedures in a stratified random sampling of skewed populations under different distributions and sample sizes. Int. J. Innov. Sci. Eng. Technol. 1 (8), 218–225.

Baumann, S., Klingauf, U., 2020. Modeling of aircraft fuel consumption using machine learning algorithms. CEAS Aeronaut. J. 11 (1), 277–287.

Chati, Y.S., Balakrishnan, H., 2016. Statistical modeling of aircraft engine fuel flow rate. In: 30th Congress of the International Council of the Aeronautical Science.

Chati, Y.S., Balakrishnan, H., 2017. A Gaussian process regression approach to model aircraft engine fuel flow rate. In: 8th International Conference on Cyber-Physical Systems. ACM, pp. 131–140.

Clarke, J.-P.B., Ho, N.T., Ren, L., Brown, J.A., Elmer, K.R., Tong, K.-O., Wat, J.K., 2004. Continuous descent approach: Design and flight test for louisville international airport. J. Aircr. 41 (5), 1054–1066.

Cuomo, S., Di Cola, V.S., Giampaolo, F., Rozza, G., Raissi, M., Piccialli, F., 2022. Scientific machine learning through physics–informed neural networks: Where we are and what's next. J. Sci. Comput. 92 (3), 88.

Dalmau, R., Prats, X., 2015. Fuel and time savings by flying continuous cruise climbs: Estimating the benefit pools for maximum range operations. Transp. Res. Part D: Transp. Environ. 35, 62–71.

Dalmau Codina, R., Melgosa Farrés, M., Prats Menéndez, X., 2018. pyBADA: Easy BADA integration in python for rapid prototyping. In: SESAR Innovation Days 2018: Posters Abstracts. pp. 16–17.

Ding, Z., Fu, Y., 2017. Deep domain generalization with structured low-rank constraint. IEEE Trans. Image Process. 27 (1), 304–313.

Gössling, S., Humpe, A., 2020. The global scale, distribution and growth of aviation: Implications for climate change. Glob. Environ. Chang. 65, 102194.

Huang, Z., Wang, H., Xing, E.P., Huang, D., 2020. Self-challenging improves cross-domain generalization. In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16. Springer, pp. 124–140.

Ilse, M., Tomczak, J.M., Louizos, C., Welling, M., 2020. Diva: Domain invariant variational autoencoders. In: Medical Imaging with Deep Learning. PMLR, pp. 322–348.

Jarry, G., Delahaye, D., 2021. Toward novel environmental impact assessment for ANSPs using machine learning. In: Climate Change and the Role of Air Traffic Control Research Workshop, Vilnius.

Jarry, G., Delahaye, D., Feron, E., 2020. Approach and landing aircraft on-board parameters estimation with lstm networks. In: 2020 International Conference on Artificial Intelligence and Data Analytics for Air Transportation. AIDA-AT, IEEE, pp. 1–6.

Jarry, G., Delahaye, D., Hurter, C., 2024a. Towards aircraft generic quick access recorder fuel flow regression models for ADS-B data. In: International Conference on Research in Air Transportation.

Jarry, G., Very, P., Dalmau, R., Sun, J., 2024b. DeepEnv: Python library for aircraft environmental impact assessment using deep learning. http://dx.doi.org/10.5281/zenodo.13754838, https://github.com/eurocontrol-asu/DeepEnv.

Kayaalp, K., Metlek, S., Ekici, S., Şöhret, Y., 2021. Developing a model for prediction of the combustion performance and emissions of a turboprop engine using the long short-term memory method. Fuel 302, 121202.

Khosla, A., Zhou, T., Malisiewicz, T., Efros, A.A., Torralba, A., 2012. Undoing the damage of dataset bias. In: Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7-13, 2012, Proceedings, Part I 12. Springer, pp. 158–171.

Lee, D.S., Fahey, D.W., Skowron, A., Allen, M.R., Burkhardt, U., Chen, Q., Doherty, S.J., Freeman, S., Forster, P.M., Fuglestvedt, J., et al., 2021. The contribution of global aviation to anthropogenic climate forcing for 2000 to 2018. Atmos. Environ. 244, 117834.

Li, H., Pan, S.J., Wang, S., Kot, A.C., 2018a. Domain generalization with adversarial feature learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5400–5409.

Li, Y., Tian, X., Gong, M., Liu, Y., Liu, T., Zhang, K., Tao, D., 2018b. Deep domain generalization via conditional invariant adversarial networks. In: Proceedings of the European Conference on Computer Vision. ECCV, pp. 624–639.
Li, H., Wang, Y., Wan, R., Wang, S., Li, T.-Q., Kot, A., 2020. Domain generalization for medical imaging classification with linear-dependency regularization. Adv. Neural Inf. Process. Syst. 33, 3118–3129.
Li, L., Yuan, S., Teng, Y., Shao, J., 2021. A study on sustainable consumption of fuel—An estimation method of aircraft. Energies 14 (22), 7559.
Liu, Q., Dou, Q., Yu, L., Heng, P.A., 2020. MS-Net: multi-site network for improving prostate segmentation with heterogeneous MRI data. IEEE Trans. Med. Imaging 39 (9), 2713–2724.
Metlek, S., 2023. A new proposal for the prediction of an aircraft engine fuel consumption: a novel CNN-BiLSTM deep neural network model. Aircr. Eng. Aerosp. Technol. 95 (5), 838–848.
Motiian, S., Piccirilli, M., Adjeroh, D.A., Doretto, G., 2017. Unified deep supervised domain adaptation and generalization. In: IEEE International Conference on Computer Vision. pp. 5715–5725.
Muandet, K., Balduzzi, D., Schölkopf, B., 2013. Domain generalization via invariant feature representation. In: International Conference on Machine Learning. PMLR, pp. 10–18.
Nuic, A., Mouillet, V., 2016. User Manual for the Base of Aircraft Data (BADA) Revision 4. EEC Technical/Scientic Report No. 12/11/22-58 v1.3, EUROCONTROL Experimental Centre.
Nuic, A., Poles, D., Mouillet, V., 2010. BADA: An advanced aircraft performance model for present and future ATM systems. Internat. J. Adapt. Control Signal Process. 24 (10), 850–866.
Olive, X., 2019. Traffic, a toolbox for processing and analysing air traffic data. J. Open Source Softw. 4, 1518.
Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al., 2011. Scikit-learn: Machine learning in Python. J. Mach. Learn. Res. 12, 2825–2830.
Poll, D., Schumann, U., 2021a. An estimation method for the fuel burn and other performance characteristics of civil transport aircraft during cruise: part 2, determining the aircraft's characteristic parameters. Aeronaut. J. 125 (1284), 296–340.
Poll, D., Schumann, U., 2021b. An estimation method for the fuel burn and other performance characteristics of civil transport aircraft in the cruise. Part 1 fundamental quantities and governing relations for a general atmosphere. Aeronaut. J. 125 (1284), 257–295.
Raissi, M., Perdikaris, P., Karniadakis, G.E., 2019. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. J. Comput. Phys. 378, 686–707.
Schäfer, M., Strohmeier, M., Lenders, V., Martinovic, I., Wilhelm, M., 2014. Bringing up OpenSky: A large-scale ADS-B sensor network for research. In: IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks. IEEE, pp. 83–94.
SESAR, 2015. European ATM Master Plan. The Roadmap for Delivering High Performing Aviation for Europe. Tech. Rep, Brussels (Belgium).
Singh, R., Mangat, N.S., Singh, R., Mangat, N.S., 1996. Stratified sampling. In: Elements of Survey Sampling. Springer, pp. 102–144.
Steininger, M., Kobs, K., Davidson, P., Krause, A., Hotho, A., 2021. Density-based weighting for imbalanced regression. Mach. Learn. 110, 2187–2211.
Sun, J., 2022. Openap. top: Open flight trajectory optimization for air transport and sustainability research. Aerospace 9 (7), 383.
Sun, J., Hoekstra, J.M., Ellerbroek, J., 2020. OpenAP: An open-source aircraft performance model for air transportation studies and simulations. Aerospace 7 (8), 104.
Sun, J., Roosenbrand, E., 2023. Fast contrail estimation with OpenSky data. In: Proceedings of 10th OpenSky Symposium. Journal of Open Aviation Science.
Tenenbaum, J.B., Silva, V.d., Langford, J.C., 2000. A global geometric framework for nonlinear dimensionality reduction. Sci. 290 (5500), 2319–2323.
Trani, A.A., Wing-Ho, F., Schilling, G., Baik, H., Seshadri, A., 2004. A neural network model to estimate aircraft fuel consumption.
Uzun, M., Demirezen, M.U., Inalhan, G., 2021. Physics guided deep learning for data-driven aircraft fuel consumption modeling. Aerospace 8 (2), 44.
Xu, Z., Liu, D., Yang, J., Raffel, C., Niethammer, M., 2020. Robust and generalizable visual representation learning via random convolutions. arXiv preprint arXiv:2007.13003.
Zhou, K., Liu, Z., Qiao, Y., Xiang, T., Loy, C.C., 2022. Domain generalization: A survey. IEEE Trans. Pattern Anal. Mach. Intell. 45 (4), 4396–4415.
Zhou, K., Yang, Y., Qiao, Y., Xiang, T., 2021. Domain generalization with mixstyle. arXiv preprint arXiv:2104.02008.