Data-driven estimation of flights' hidden parameters

George A. Vouros, Theodore Tranos, Konstantinos Blekas, Georgios Santipantakis
University of Piraeus Research Center Piraeus, Greece
georgev@unipi.gr

Marc Melgosa, Xavier Prats
Universitat Politècnica de Catalunya - Barcelona Tech
Barcelona, Spain
xavier.prats@upc.edu

Abstract— **This paper presents a data-driven methodology for the estimation of flights' hidden parameters, combining mechanistic and AI/ML models. In the context of this methodology the paper studies several AI/ML methods and reports on evaluation results for estimating hidden parameters, in terms of mean absolute error. In addition to the estimation of hidden parameters themselves, this paper examines how these estimations affect the prediction of KPIs regarding the efficiency of flights using a mechanistic model. Results show the accuracy of the proposed methods and the benefits of the proposed methodology. Indeed, the results show significant advances of data-driven methods to estimate hidden parameters towards predicting KPIs.**

Keywords: Hidden parameters; data-driven estimation; KPIs; prediction; AI/ML

I. Introduction

The development of performance modelling methodologies able to grasp the interdependencies between different Key Performance Areas (KPAs) and translate new ATM concepts and technologies into their impact on high-level, system-wide Key Performance Indicators (KPIs) has been a long-time objective of the ATM research community. Generally speaking, the modelling approaches to this problem can be classified into two main categories: macroscopic and microscopic. Macroscopic models represent the behaviour of a system by formulating the relationships between aggregated variables without explicitly modelling the individual system components. On the contrary, microscopic models adopt an explicit representation of the actions and interactions of the individual elements that compose a system with the aim to observe the performance that emerges at the macroscopic level. Several SESAR ER projects, such as APACHE[1], Vista[2] and EvoATM[3], have employed different types of microscopic models (e.g., agent-based models) to understand the influence of new ATM concepts and solutions on the performance of the system as a whole, showing the ability of these models to capture a rich variety of behaviours in a very realistic manner. However, the practical application of complex simulation models to strategic ATM performance assessment and decision-making is hindered by several factors. One of the most important is hidden parameters: even if they could, in principle, be measured, certain aspects of the ATM system may not be observable for practical reasons. This is the case, for example, of business sensitive data related to the behaviour of airspace users (AUs) that are of paramount importance for the construction of microsimulation models, such as aircraft take-off weight (TOW), selected cost index (CI), payload mass (PL), etc.

In the last decade, with the rising interest in artificial intelligence, transport and traffic modelers have begun to apply a variety of machine learning techniques for hidden parameter estimation that are proving successful in improving the capabilities of microsimulation models ([1], [2], [3]). However, the exploration of these techniques in the field of ATM is only very recent. A common approach in many simulation exercises is to set these parameters based on some typical values recommended in the literature, due to the difficulties to conduct a rigorous and systematic calibration on these unknown parameters ([4], [5]). However, these parameters can differ significantly across AUs. The question, as in any calibration exercise, is essentially how to explore the parameter space of each AU model in order to find the combination of parameter values that better matches the observed trajectory choices of that AU. The problem is therefore similar to that of efficiently exploring the model input-output space, with the difference that, in this case, the goal is not to optimize a certain performance function or find combinations of inputs leading to a particular outcome, but to minimize the difference between the simulated trajectories and the trajectories observed in historical data. This problem has been addressed in SIMBAD [16] building on recent work ([6]; [7]) and reported in this article.

The main objective of this research is to explore the use of machine learning techniques for the estimation of flights' hidden parameters from historical data. Specifically, this paper presents a data-driven methodology for the estimation of flights' hidden parameters, combining mechanistic and AI/ML models. While mechanistic models produce optimal trajectories with respect to specific hidden parameters values, data-driven methods are trained to predict the target hidden parameters' values, given the trajectories. In the context of this methodology the paper proposes and evaluates several AI/ML methods to estimate payload mass (PL) and cost index (CI), and reports on evaluation results in terms of mean absolute error (MAE). In addition to the estimation of hidden parameters themselves, this paper examines how these estimations affect the prediction of KPIs regarding the efficiency of flights: fuel consumption, gate-to-gate time and distance flown. Results show the accuracy of the proposed methods and the benefits of the proposed methodology.

[1] www.sesarju.eu/projects/apache
[2] https://www.sesarju.eu/projects/vista
[3] www.sesarju.eu/projects/evoATM

The contributions that this paper makes are the following:

- First, it provides a specific methodology for the data-driven estimation of hidden parameters, using mechanistic models for the provision of training examples.
- Second, a set of AI/ML methods are tuned and evaluated for the estimation of hidden parameters, providing a set of comprehensive, comparative results.
- Third, data-driven AI/ML models are evaluated in the context of the overall methodology, where specific flights' KPIs are predicted. This shows how the (in-)accuracy of estimations affects the prediction of specific KPIs.

The structure of this paper is as follows: Section II presents the overall methodology to train the AI/ML models for the estimation of hidden parameters and the prediction of KPIs. Section III formulates the data-driven estimation of hidden parameters as a regression problem, and describes the datasets exploited for training/testing the AI/ML models. Section IV describes the AI/ML methods used for the estimation of hidden parameters and section V presents the results of these methods. Additionally, Section V shows how hidden parameters estimated by the best AI/ML method affect the prediction of specific KPIs. Section VII concludes the paper with final remarks and future work.

II. Methodology

Fig. 1 specifies the overall methodology for the estimation of hidden parameters and prediction of flights' KPIs. It specifies the pipelines with components for the provision of training/testing data, AI/ML methods, as well as components for the prediction of KPIs.

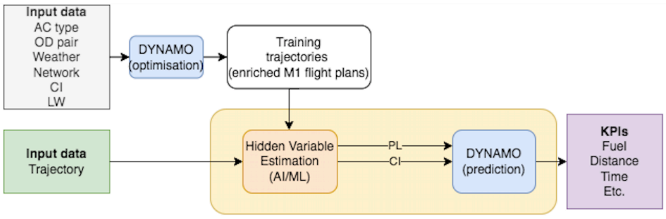

Figure 1. Overall data-driven methodology for estimation of hidden parameters (training and evaluation)

Specifically, Fig. 1 shows two pipelines: One for the training of AI/ML models (top) and one (bottom) for using the models for estimating hidden parameters. The later concludes with the prediction of KPIs. These pipelines comprise the DYNAMO component in optimization and prediction modes.

DYNAMO [8] is an aircraft trajectory prediction and optimization engine capable to rapidly compute trajectories using realistic and accurate weather, and aircraft performance data. DYNAMO is based on an aircraft point-mass model (3 degrees of freedom) and its design enables it to be used on real-time applications and/or when a large set of trajectories needs to be rapidly generated for simulation or benchmarking purposes. Moreover, DYNAMO is highly flexible and configurable and allows the user to specify a great variety of constraints and objective functions.

In the context of the proposed methodology, DYNAMO is used in order to (a) create training/testing datasets to train/evaluate the AI/ML models as shown in the first (upper) pipeline, and (b) predict flights' KPIs in a mechanistic way, as shown the second (lower) pipeline.

Operating in optimization mode, $\text{DYNAMO}_{\text{optimization}}$ provides flight plans that are enriched with some variables (specified subsequently), in conjunction to the values of the hidden parameters CI and PL, per trajectory. These trajectories and target values are used for training the AI/ML models.

Flight plans provided by $\text{DYNAMO}_{\text{optimization}}$ are called DYNAMO_FP trajectories. The input variables for $\text{DYNAMO}_{\text{optimization}}$ are the following ones:

- Weather data: GRIB file (wind, pressure, temperature).
- CI.
- Take-off mass or landing mass or PL.
- Aircraft type.
- Airspace structure (free route areas, entry/exit points, airways in non-free route airways...).
- Initial trajectory (if the lateral route has to be fixed totally or partially).
- Route charges (if flying in Europe).
- Origin – destination (OD) pair.

The second pipeline takes as input a trajectory (flight plan) and uses the well-trained AI/ML model to estimate the hidden parameters' values. This time DYNAMO operates in prediction mode ($\text{DYNAMO}_{\text{prediction}}$), to estimate for the given trajectory the target KPIs, given the estimated hidden parameters.

The datasets exploited from these pipelines are:

- Weather Conditions data.
- Flight plans.
- Simulated flight data.

The first dataset provides weather conditions on a predefined spatial grid of fixed positions and time intervals. The data have been retrieved from the Copernicus Climate Change Service (C3S) at ECMWF, covering 28 days in the period from January 2018 to December 2018. This data source provides hourly estimates of a large number of atmospheric, land and oceanic climate variables. The data cover the Earth on a 30km grid and resolve the atmosphere using 137 levels from the surface up to a height of 80km.

For the second dataset (flight plans), EUROCONTROL provides to operational stakeholders an accurate picture of past and future air traffic demand over the European continent via the Demand Data Repository (DDR2). The flight plans in this dataset are provided in the ALLFT+ format (version 4). Each record in an ALLFT+ file reports the flight plan of a single flight in 181 columns. Each flight plan comprises at most three profiles: a) Filed Tactical Flight Model (FTFM or M1), b) Regulated Tactical Flight Model (RTFM or M2), and c) Current Tactical Flight Model (CTFM or M3). We use only the M1 flight plans. These are depicted in Fig. 2.

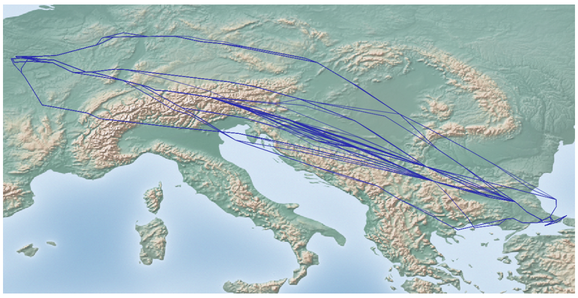

Figure 2. The M1 trajectories in the data set of flight plans

Finally, the third dataset includes DYNAMO_FP, i.e., DYNAMO$_{\text{optimization}}$ simulated trajectories of flights for the 250 possible combinations of 50 CI values (ranging from 0 to 100, with discretization interval 2) and 5 PL values (ranging from 0.6 to 1, with discretization interval 0.1). This dataset comprises 50250 files, i.e., one file for each simulated trajectory for a given combination of CI and PL values.

The DYNAMO_FP dataset reports 201 distinct flights, which connect Charles de Gaulle (LFPG) and Istanbul Ataturk airports (LTBA), corresponding to ALLFT+ M1 flight plans shown in Fig. 2, operated in any of the 28 days for which weather conditions are available, with various aircraft types.

As provided by DYNAMO, each record reports the position of the aircraft for a specific time (UTC), its altitude (both geometric and pressure altitudes), vertical and horizontal speed and throttle, the computed phase of the trajectory, as well as weather conditions for the given altitude, position and time.

Since some of the variables enriching DYNAMO_FP trajectories are not provided for real-world flight plans, the pre-processing task enriches trajectories with variables that can be computed from real-world M1 flight plans. This is done for all DYNAMO_FP trajectories and for each combination of CI and PL values provided in this dataset, ignoring all - except 3D positional - variables per trajectory point provided by DYNAMO. The detailed variables for this dataset are specified in Section III.

Data pre-processing involves mainly three major tasks. Firstly, weather conditions reported in the first data enrich the trajectory points (positions) regarding trajectories provided in the second (ALLFT+ flight plans) and trajectories provided in the third (DYNAMO_FP simulated flight data) data set. Finally, the pre-processing task is completed (a) by a rough estimation of the flight phases, according to aircraft positional data, and (b) the computation of features that are exploited for the estimation of hidden parameters.

The estimation of flight phases is a crucial pre-processing task applied to all trajectories. The phases of a flight are reduced to those in the set {climbing below FL100, climbing above FL100, cruising, descending above FL100, descending below FL100}, but we exclude phases below FL100. The recognition of flight phases is done by means of the intention of the movement of the aircraft. For example, the phase "climbing_above_FL100" for each flight, indicates the intention of the aircraft for climbing, while "cruising" phase indicates the intention of preserving its current flight level: Indicative results for estimating the phases of flights are provided in Fig. 3.

Therefore, the simulated flight plans in the DYNAMO_FP dataset results into two distinct datasets which are distinguished by the set of variables per trajectory point: one with trajectory variables provided by DYNAMO$_{\text{optimization}}$, and the other with trajectory variables provided by the pre-processing method. Both datasets are being used for training/testing the AI/ML models, although the later one is the one closer to reality. The next section specifies the features exploited per dataset for the estimation of hidden parameters.

III. Problem Formulation

The goal is the estimation of the hidden parameters, CI and PL, given a flight plan and trajectory variables (in our case, either provided by DYNAMO$_{\text{optimization}}$, or provided by the pre-processing module). This can be casted as a regression problem that targets the prediction of a vector of parameters Y, given a vector of input variables X.

In other words, the aim is to approximate a function f, such that

$$Y=f(X)+e,$$

where e constitutes a noise term or represents imperfections on data.

In our case, the vector Y is a two-dimensional output variable that corresponds to two hidden parameters, while X is a feature vector that is derived from the variables that enrich either DYNAMO_FP trajectories with X comprising 69 features (denoted as DYNAMO_FP(69)), or with 51 features derived from trajectory variables estimated by the pre-processing module (denoted as DYNAMO_FP(51)).

The variables exploited and the input features for each case are described next.

A. *Features for DYNAMO_FP(69)*

Each DYNAMO_FP simulated trajectory is split into phases (provided by DYNAMO$_{\text{optimization}}$) and, for each phase, the following features (independent variables) are considered: The median and the interquartile range (IQR, Q3 – Q1) per flight phase, for each one of the 11 variables presented in Table I (resulting into 22 features per flight phase, i.e., 66 features per trajectory), and 3 additional features for the duration of each flight phase. This results into (11 variables X 2 (median,IQR)) X 3 (phases) + 3 = 69 input features and a dependent output vector of 2 values: CI and PL.

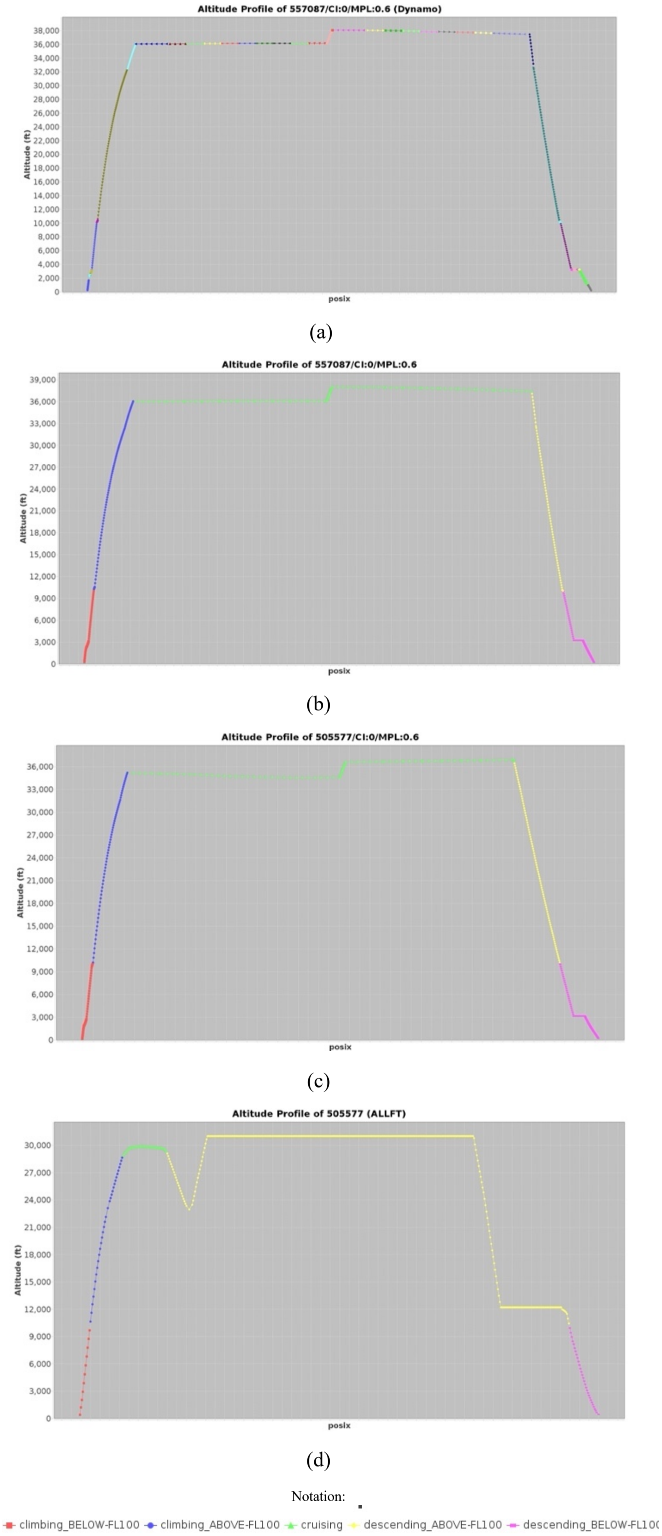

Figure 3. The phases for two indicative flights as provided by DYNAMO ((a) in a higher resolution and (c) according to the phases targeted) and as computed by the pre-processing method ((b) and (d)). The case shown in (b) is a good estimation compared to what DYNAMO specifies, but the case shown in (d) shows an incorrect estimation of flight phases compared to what is specified by DYNAMO, as shown in (c).

TABLE I. TRAJECTORY VARIABLES FROM DYNAMO$_{\text{OPTIMIZATION}}$

Variable	Description
h[ft]	geometric altitude
Temp[oC]	air temperature
Press[hPa]	air pressure
Wn[kt]	North wind component
We[kt]	East wind component
Ws[kt]	Along path wind component
Wx[kt]	Cross-wind component
Lat[o]	Latitude
Lon[o]	Longitude
vdot	Derivative of True Airspeed
hdot	Derivative of geometric altitude

B. *Features for DYNAMO_FP(51)*

Each trajectory is split into phases estimated by the pre-processing module, exploiting only spatio-temporal information of trajectories, and, for each phase, the following features are considered.

The median and the IQR per phase of each one of the 9 variables presented in Table II (resulting into 18 features per flight phase, i.e., 54 features per trajectory), and the duration of each flight phase, which results into 3 additional features.

TABLE II. TRAJECTORY VARIABLES PROVIDED BY THE PRE-PROCESSING METHOD

Variables	Description
h[ft]	geometric altitude
Temp[oC]	air temperature
v-Wn[kt]	v wind component
u-We[kt]	u wind component
Lat[o]	Latitude
Lon[o]	Longitude
vdot	Derivative of speed
hdot	Derivative of geometric altitude

IV. AI/ML METHODS

This section provides a brief description of the methods evaluated to estimate the CI and the PL.

Lasso Regression: This is a type of linear sparse regression model that is based on a sparseness procedure over the linear weights of model parameters, well-suited for models showing high levels of multicollinearity [9].

Neural Networks: An artificial neural network (NN) is a universal approximator that has the ability to learn complex non-linear relationships among data [10]. The learning process establishes an optimisation strategy using an appropriate optimisation algorithm and a loss function, which is trying to minimise. Here, the Adam optimiser [11] and the mean absolute error (MAE) loss function [12] are used.

Support Vector Regression: The use of support vector machines (defining a hyperplane, and fitting as many instances as is feasible within this hyperplane while at the same time limiting margin violations) for regression problems is known as support vector regression (SVR) [13]. SVR also uses the kernel trick by introducing an appropriate kernel function (e.g. radial basis function (RBF or sigmoidal kernel) that acts like a function approximator [10].

Kernel Ridge Regression: This method (KRR) [14] combines ridge regression (linear least squares with L2-norm regularization) with the kernel trick. It thus learns a function in the space induced by the respective kernel and the data. For non-linear kernels, this corresponds to a non-linear function in the original space.

Gradient-Boosting trees: This is a machine learning technique for optimising the predictive value of a model through successive steps in the learning process. Typically, a decision tree is used as the basic weak ML model. The gradient boosting method (GBM) works in a stage-wise manner, iteratively adding a tree model that focuses on correcting the mistakes from the previous models. In contrast, other ensemble methods train the models in isolation and this might simply lead to each model making the same mistakes.

V. EXPERIMENTS

A. *Experimental setting*

Every flight plan comprises approx. 600 points, and every such point is enriched with the trajectory variables shown in Table I for DYNAMO_FP(69) and in Table II for DYNAMO_FP(51). These variables are used for the calculation of the corresponding features, as specified in Section III.

The two trajectory datasets have been split into training and testing subsets. Given 68 ALLFT+ M1 trajectories chosen for the purposes of validating trajectory modelling methods (not described in this paper), and given all possible combinations of CI and PL values for these flights, there are approx. 15700 DYNAMO_FP trajectories that have been used to test the hidden parameters estimation methods.

The experimental study follows a 10-fold cross validation strategy from which the best hyperparameters for all the different models are obtained, while the loss function considered is MAE. Models for DYNAMO_FP(69) and DYNAMO_FP(51) data sets have been tuned separately, given that these have different trajectory variables. This also serves the purpose of computing the difference between estimated parameters using (a) enriched flight plans, with all the variables provided by optimization methods, against (b) real-world flight plans whose variables are approximated by pre-processing methods.

The input features were scaled to the [0,1] interval, while the estimated hidden parameters are left unscaled. Experiments showed that scaling CI and PL made no difference in estimating the hidden parameters, while scaling the input was mandatory for almost all methods. Finally, the models' estimations were rounded to the closest integer for CI and to the 1st decimal position for PL.

The control variables' values per method are specified in Table III and Table IV, for DYNAMO_FP(69) and for DYNAMO_FP(51), respectively. Parameters of the machine learning methods for the DYNAMO(69) dataset

B. *Experimental results for DYNAMO(69)*

The experimental results of all methods in the test set of DYNAMO_FP(69) trajectories are presented in Table V. In particular, this table shows the performance of the five methods in terms of MAE in all DYNAMO_FP test trajectories. These errors are calculated after executing 1 run of the best model per method in the test subset of DYNAMO_FP trajectories.

Specifically, results report the following quantities:

- the MAE mean value (mean),
- the MAE standard deviation (std),
- the interquartile MAE range IQR = Q3 – Q1,
- the MAE range (max – min).

TABLE III. ARAMETERS OF THE MACHINE LEARNING METHODS FOR THE DYNAMO(69) DATASET

Method	Settings
LASSO	$\lambda = 10^{-6}$
NN	Number of hidden layers = 2 Number of neurons = 100 and 50 respectively Learning rate = 10^{-4} Weight Decay = 0 Epoch s= 50.000 Activation function = linear
SVR	Kernel function = RBF C = 0.25 $\epsilon = 10^{-2}$ tolerance = 10^{-3}
KRR	Kernel function = RBF $\alpha = 10^{-2}$ gamma (Parameter for the RBF kernel) = 2×10^{-2}
GBM	n_estimators = 1000 min_samples_leaf = 15 min_samples_split = 10 max_depth = 8 learning_rate = 10^{-1}

TABLE IV. PARAMETERS OF THE MACHINE LEARNING METHODS FOR THE DYNAMO(51) DATASET

Method	Settings
LASSO	$\lambda = 10^{-7}$
NN	Number of hidden layers = 2 Number of neurons = 100 and 100 respectively Learning rate = 10^{-4} Weight Decay = 10^{-6} Epochs = 50.000 Activation function = linear
SVR	Kernel function = RBF C = 1 $\epsilon = 10^{-3}$ tolerance = 10^{-3}
KRR	Kernel function = RBF $\alpha = 5 \times 10^{-3}$ gamma (Parameter for the RBF kernel) = 10^{-2}
GBM	n_estimators = 1000 min_samples_leaf = 10 min_samples_split = 15 max_depth = 8 learning_rate = 10^{-1}

The results are also shown in Fig. 4 using boxplots. This figure shows the MAE values (y-axis) for the CI variable (left) and PL variable (right) with each AI/ML method used, as indicated in the x-axis.

TABLE V. PERFORMANCE OF ALL REGRESSION METHODS IN DYNAMO_FP(69)

Method	CI MAE				PL MAE			
	mean	std	IQR	range (max - min)	mean	std	IQR	range (max - min)
GBM	2.91	3.59	4	28	0.009	0.032	~0	0.3
NN	3.89	3.89	5	30	0.016	0.044	~0	0.3
KRR	4.06	4.322	5	42	0.026	0.048	~0	0.2
SVR	3.94	3.97	5	34	0.021	0.049	~0	0.3
LASSO	6.032	5.04	7	46	0.039	0.055	0.01	0.3

C. *Experimental results for DYNAMO(51)*

The experimental results of all methods using the model and the test set of DYNAMO_FP(51) trajectories are presented in Table VI. In particular, this table shows the performance of the five methods in terms of MAE per hidden parameter. These errors are calculated after executing 1 run of the best model in the test subset of DYNAMO_FP(51) trajectories. Table VI reports on the same quantities reported for the DYNAMO(69) dataset.

The same results are also shown in Fig. 5 using boxplots. Similar to Fig. 4, this figure shows the MAE values for the CI hidden variable (left) and the PL hidden variable (right) obtained with each AI/ML method tested.

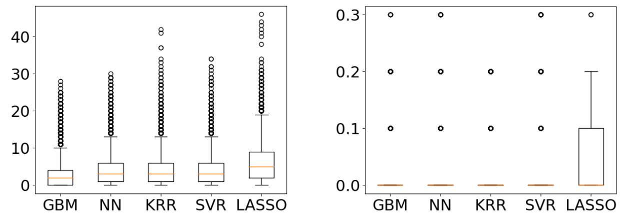

Figure 4. Boxplots of results for DYNAMO(69). The Y axis corresponds to the MAE of hidden parameters estimation (left: CI, right: PL) and the X axis indicates the ML method used

According to the results, the best mean performance for CI is obtained using the GBM method. However, GBM in this case reports a large range of values for CI compared to NN, with the same std. Additionally, GBM reports a smaller range of values compared to SVR which is the 2nd best method. Considering PL, the best method is KRR with a slight difference from GBM.

TABLE VI. PERFORMANCE OF ALL REGRESSION METHODS IN DYNAMO_FP(51)

Method	CI MAE				PL MAE			
	mean	std	IQR	range (max - min)	mean	std	IQR	range (max - min)
GBM	3.65	4.2	4	37	0.022	0.046	~0	0.3
NN	4.38	4.21	5	34	0.023	0.046	0	0.3
KRR	4.05	3.86	5	44	0.021	0.046	0	0.3
SVR	3.98	4.21	5	41	0.03	0.052	0.1	0.4
LASSO	8.06	6.49	9	88	0.078	0.068	0.1	0.3

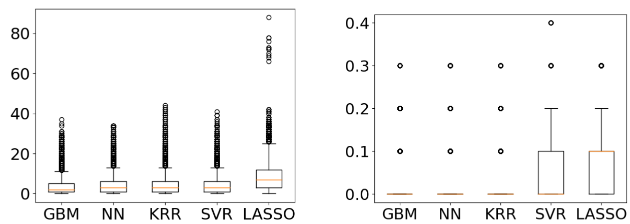

Figure 5. Boxplots of results for DYNAMO(51). The Y axis corresponds to the MAE of hidden parameters estimation (left: CI, right: PL) and the X axis indicates the ML method used

Thus, the GBM method has significantly better results for CI compared to the other methods, and slightly worse than KRR for PL. KRR is the third best model for estimating CI. Therefore, GBM provides the best balance in estimating both hidden parameters compared to the other methods, achieving a MAE below 4% for CI and approx. 2% for PL.

Overall, from the results obtained with all the hidden parameters estimation methods used, we can conclude that GBM achieves the best balance for estimating CI and PL in all cases, with a small std (thus, exhibiting robustness and stability) and a small estimation error. This concludes that GBM is the best among all methods and the mean absolute error reported is (a) for CI less than two times the discretization interval of the CI values provided in the training data (actually, 1.8 of the discretization interval, which is equal to 2), or, less than 4% (i.e., 4 units in [0,100]), and (b) for PL is much less than the discretization interval (equal to 0.1) of the values provided in the training data, or, less than 2% (i.e., 0.02 units in [0,1]).

Also, the results show that all methods show consistent behavior when trained and tested using the features provided by DYNAMO, and when trained and tested in approximations provided by the pre-processing methods. This is exactly the case for GBM, which performs consistently well in both cases, and for both hidden parameters, compared to the other methods evaluated. This shows that GBM can be used for the estimation of hidden parameters for real-world flight plans whose variables are provided by pre-processing methods.

D. The effect of hidden parameters' estimation on the prediction of KPIs

This part reports on how the estimations of hidden parameters provided by the best model trained (i.e., GBM), for M1 trajectories, support the prediction of flights' KPIs using standard prediction error metrics (such as MAE). The KPIs are: fuel consumption, flown distance, and gate-to-gate time. Here we provide results for the DYNAMO_FP(51) flight plans with the known hidden parameters' values and with the estimated ones. As the true values of the hidden parameters are known in this case, the aim is to study how the difference between known and estimated values of the hidden parameters is reflected in differences between predicted and actual KPIs provided by mechanistic models. KPIs are computed using $\text{DYNAMO}_{\text{prediction}}$.

The predicted KPIs are provided by $\text{DYNAMO}_{\text{prediction}}$ considering the estimated values of hidden parameters, and the actual KPIs are provided by $\text{DYNAMO}_{\text{prediction}}$ considering the known values of hidden parameters per trajectory. Hence, the comparison is between KPIs predicted using estimated hidden parameters and KPIs predicted using the true hidden parameters, both provided by $\text{DYNAMO}_{\text{prediction}}$. For that, as said, the best hidden parameters prediction model, i.e., GBM, is used to estimate the hidden parameters of the DYNAMO_FP test trajectories whose hidden parameters' true values are also provided. We consider the DYNAMO_FP(51) case only, since this is closer to the real-world case of model deployment. The overall comparison process described is shown in Fig. 6.

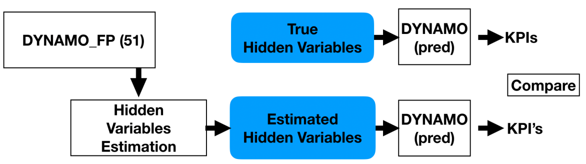

Figure 6. The overall process for estimating the effect of hidden parameters' estimation errors on the prediction of KPIs

Abs difference on predicted CI	Abs difference on predicted PL: 0	0.1	0.2	0.3
0	0 (17.79%)	1.93 (1.48%)	3.47 (0.08%)	N/A
1	0 (17.67%)	1.98 (2.56%)	3.73 (0.15%)	4.55 (0.02%)
2	0 (10.43%)	1.95 (2.31%)	3.74 (0.14%)	N/A
3	0 (7.81%)	1.91 (2.15%)	3.91 (0.15%)	N/A
4	0 (6.42%)	2.03 (2.02%)	3.70 (0.11%)	5.73 (0.01%)
5	0 (4.37%)	1.99 (1.53%)	3.64 (0.15%)	N/A
6	0 (3.29%)	2.05 (1.28%)	3.41 (0.12%)	7.40 (0.01%)
7	0 (2.52%)	1.99 (1.09%)	3.93 (0.08%)	N/A
8	0 (2.02%)	1.93 (0.98%)	3.16 (0.09%)	7.36 (0.01%)
9	0 (1.48%)	1.92 (0.76%)	3.72 (0.05%)	7.41 (0.01%)
10	0 (1.09%)	2.03 (0.59%)	3.72 (0.07%)	N/A
11	0 (0.92%)	1.93 (0.52%)	3.50 (0.08%)	7.43 (0.01%)
12	0 (0.57%)	1.97 (0.53%)	3.63 (0.05%)	7.34 (0.01%)
13	0 (0.44%)	1.95 (0.32%)	3.77 (0.07%)	N/A
14	0 (0.41%)	1.94 (0.31%)	4.11 (0.07%)	N/A
15	0 (0.33%)	1.79 (0.20%)	4.02 (0.07%)	N/A
16	0 (0.29%)	1.74 (0.15%)	3.61 (0.01%)	N/A
17	0 (0.15%)	1.83 (0.14%)	4.35 (0.01%)	N/A
18	0 (0.18%)	1.68 (0.10%)	3.54 (0.01%)	N/A
19	0 (0.14%)	1.78 (0.06%)	N/A	N/A
20	0 (0.11%)	1.80 (0.09%)	4.42 (0.01%)	N/A
21	0 (0.06%)	1.52 (0.06%)	3.24 (0.01%)	N/A
22	0 (0.04%)	1.44 (0.07%)	4.19 (0.01%)	N/A
23	0 (0.07%)	1.66 (0.04%)	N/A	N/A
24	0 (0.04%)	1.75 (0.04%)	4.12 (0.01%)	N/A
25	0 (0.03%)	1.41 (0.04%)	4.90 (0.01%)	N/A
26	0 (0.04%)	1.48 (0.03%)	3.90 (0.02%)	N/A
27	0 (0.04%)	1.59 (0.01%)	3.95 (0.01%)	7.39 (0.01%)
28	0 (0.02%)	1.43 (0.02%)	3.96 (0.02%)	N/A

Figure 7. The effect of hidden parameters' estimation errors on the prediction of Fuel [kg]: Rows correspond to absolute differences on estimating CI, columns correspond to absolute differences on estimating PL, and each cell indicates two values: (a) the MAPE of predicted vs actual fuel for all trajectories with the corresponding differences on CI and PL, and (b) the percentages of trajectories in those CI & PL differences. N/A means that there where not cases with this combination of CI and PL absolute differences.

Given that $\text{DYNAMO}_{\text{prediction}}$ predictions for flown distance and gate-to-gate time do not differ when $\text{DYNAMO}_{\text{prediction}}$ considers the true or the predicted estimated values of parameters, below we report only differences regarding the fuel consumption prediction.

Fig. 7 shows the effect of hidden parameters estimation errors on the predictions of fuel consumption. Specifically, Fig. 7 rows correspond to absolute differences on estimating CI, columns correspond to absolute differences on estimating PL, and the values in cells show the mean absolute percentage of error (MAPE) in the fuel consumption prediction for all the trajectories with the corresponding differences on CI and PL. In addition, the percentage of testing cases are reported in parentheses in each cell. Results show that both hidden parameters play conjunctively a role on increasing the error of fuel consumption estimation. However, PL may play a more critical role: large errors in PL may be translated to large differences in predicted fuel consumption. It is interesting to note that more than 78% of all test cases have no error (1st column of Table presented in Fig. 7), while the average MAPE value of all cases is 0.45 %.

Furthermore, as Fig. 8 shows, the distribution of fuel consumption in all cases, either with the estimated hidden parameters or with the true hidden parameters, are the same, and this happens (following a t-test on these predictions) with probability 0.99.

This verifies the efficacy of the hidden parameters' estimation methods to support the prediction of trajectory KPIs. The MAPE for fuel is below 1%, while for distance and gate-to-gate-time is 0%. Therefore, we can conclude that estimated hidden parameters for known trajectories support the prediction of trajectory KPIs with high accuracy.

This result provides firm evidence to the hypothesis that the hidden parameters estimation errors reported have insignificant consequences to the prediction of KPIs.

VI. CONCLUSIONS

This paper explores the use of machine learning techniques for the estimation of flights' hidden parameters from historical data. Specifically, it presents a data-driven methodology for the estimation of flights' hidden parameters, combining mechanistic and AI/ML models. In the context of this methodology the paper proposes and evaluates several AI/ML methods to estimate PL and CI. In addition, the paper examines how these estimations affect the prediction of KPIs related to flight efficiency: fuel consumption, gate-to-gate time and distance flown. The results show the accuracy of the proposed methods and the benefits of the proposed methodology.

Future work aims to explore deep AI/ML methods for the estimation of hidden parameters, also expanding the set of parameters, with the aim to further increase the accuracy of estimations and the accuracy of KPI predictions.

ACKNOWLEDGMENT

This work has received funding from SESAR Joint Undertaking (JU) within SIMBAD project under grant agreement No 894241. The JU receives support from the European Union's Horizon 2020 research and innovation programme and the SESAR JU members other than the Union. Also, we thank ENAIRE's operational staff for their support in the use cases validation.

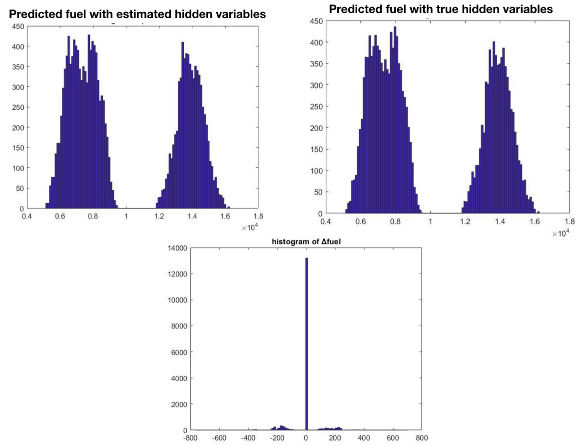

Figure 8. The distributions of predicted fuel consumption[kg] given the estimations of hidden parameters (top left), and the true hidden parameters (top right), as well as the distribution of the absolute difference in the predicted fuel (bottom).

REFERENCES

[1] Feng, T. and H.J. Timmermans (2016). "Comparison of advanced imputation algorithms for detection of transportation mode and activity episode using GPS data", Transp. Planning and Tech., 39(2), 180-194.

[2] Hernández, Y., T. Djukic and J. Casas (2018). "Local traffic patterns extraction with network-wide consistency in large urban networks", Transportation Research Procedia, 34, 259-266

[3] Antunes, F., M. Amorim, F.C. Pereira and B. Ribeiro (2019). "Active learning metamodeling for policy analysis: application to an emergency medical service simulator", Simulation Modelling Practice and Theory, 97, 101947.

[4] COPTRA Consortium (2017). COPTRA Deliverable D2.1: Techniques to determine trajectory uncertainty and modelling, Edition 01.00.00.

[5] Dalmau, R., Melgosa, M., Vilardaga S. and Prats, X. 2018a. A Fast and Flexible Aircraft Trajectory Predictor and Optimiser for ATM Research Applications, Proceedings of the 8th ICRAT.

[6] Chati, Y. S. and Balakrishnan, H. (2017). "Statistical modeling of aircraft take-off weight", 12th USA/Europe Air Traffic Management Research and Development Seminar, Berkeley, California, USA.

[7] Sun, J., J. Ellerbroek and J. Hoekstra (2017). "Bayesian inference of aircraft initial mass", 12th USA/Europe Air Traffic Management Research and Development Seminar, Berkeley, California, USA

[8] R. Dalmau, M. Melgosa, S. Vilargada and X. Prats, "A fast and flexible aircraft trajectory predictor and optimiser for ATM research applications," in *ICRAT*, Castelldefels, Catalonia, 2018.

[9] R. Tibshirani, Regression Shrinkage and Selection via the Lasso, Journal of the Royal Statistical Society, 1996.

[10] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006

[11] J. D.P. Kingma, "Adam: A Method for Stochastic Optimization," arxiv:1412.6980, 2014.

[12] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014.

[13] B. E. Boser, I. M. Guyon and V. N. Vapnik, A Training Algorithm for Optimal Margin Classifiers, ACM Press, 1992.

[14] V. Vovk, Kernel Ridge Regression, Springer, 2013.

[15] T. G. Dietterich, "Ensemble Methods in Machine Learning," in *Multiple Classifier Systems*, Springer, 2000.

[16] SIMBAD «Combining Simulation Models and Big Data Analytics for ATM Performance Analysis", https://www.sesarju.eu/projects/SIMBAD