# Learning structural stress virtual sensors from on-board instrumentation of a commercial aircraft

Martin Ghienne, Alexandre Limare

# Learning structural stress virtual sensors from on-board instrumentation of a commercial aircraft

Martin Ghienne[a,*], Alexandre Limare[a]

[a]*Institut supérieur de mécanique de Paris (ISAE-Supméca), Laboratoire Quartz, 3 rue Fernand Hainaut, 93407 Saint-Ouen-sur-seine cedex, France.*

**Abstract**

This work aims to predict the mechanical stress on the structure of a business jet in service phase from flight instrument only. A significant database obtained from test flights using aircraft instrumented with strain gauges has been provided as part of an Artificial Intelligence challenge organised by the french Ile-de-France region and the aircraft manufacturer Dassault Aviation. Learning techniques are considered to train a prediction model of the aircraft structural stress. The proposed baseline includes a clustering step for phase identification in time series and an ensemble model with two stacked regressors. The model is trained on a dataset of 117 flights and its overall performance is evaluated on a validation set of flight sequences from 186 flights. The main advantages of the proposed learning approach are its prediction accuracy, its training frugality and its interpretability. This paper presents a global data science workflow applied to a problem of structural stress prediction. Despite a development in a constrained time period with no direct access to the data, the proposed approach demonstrates the feasibility of the concept of learned virtual sensor of aircraft structural stress and paves the way for applications to other structures.

*Keywords:* Structural stress virtual sensors, Time series regression, ensemble model learning, Challenge AI for Industry, Aircraft Digital Twin

**Contents**



*Corresponding author

*Email addresses:* martin.ghienne@isae-supmeca.fr (Martin Ghienne), alexandre.limare@isae-supmeca.fr (Alexandre Limare)

## 1. Introduction

Informally introduced in 2002 by Michael Grieves (formalized later in his white paper [1]), the Digital Twin (DT) concept provides the opportunity to optimize a product or a system through all phases of its life cycle. Indeed, by fulfilling the gap between the physical world to the digital world, the DT constitutes a single reference point for design, engineering, manufacturing and service phases [2]. As pointed out by Liu et al [3], DT relies on key technologies which can be categorized into data related technologies, high-fidelity modeling technologies, and model based simulation technologies. Data related technologies include data collection, data mapping, data processing and data transmission while high-fidelity modeling includes physics modeling, semantic modeling and model integration. Model based simulation technologies refer, among other, to multi-scale simulation, finite element analysis or discrete event simulation. Despite the emergence of the DT concept almost twenty years ago, the actual implementation of this concept is becoming possible only recently thanks to the advances in Artificial Intelligence (AI), big data analytics, Internet-of-Things(IoT), cloud computing and High Performances Computing (HPC) technologies.

The interest for the DT Concept has been growing rapidly in many industrial sectors and more recently in healthcare sector where physiological DT and hospital management optimization are of particular interest. The aviation sector was one of the first to take an interest in the development of DT, with the purpose of predictive maintenance, decision support, optimization, and diagnostics [2]. Tuegel et al. studied the feasibility of an Airframe Digital Twin (ADT) [4, 5]. They focused on the modeling requirements needed to perform the ADT. In the same time, Gockel et al. developed a rudimentary DT of an aircraft using Computation Fluid Dynamic (CFD) model and Finite Element model. Flight data are used as input of the CFD model which generates aerodynamic flight loads applied to the FE model. They identify technology gaps to be addressed in order to complete the achievement of an actual DT. Among the technical obstacles identified by the authors, several issues related to CFD simulations highlight the difficulty to evaluate structural loads from numerical simulation. Li et al. [6] proposed a DT based on a Dynamic Bayesian Network for the Structural Health Monitoring (SHM) of an aircraft wing. It allows to diagnose and forecast damages on the wing considering a simple FE model of the wing, loads measured from sensors in the wing. The major benefit of the proposed DT is that it integrates various uncertainty sources of the problem. Focusing more on the exploitation of the DT than on its implementation, Millwater et al. [7] review current probabilistic approaches and their potential synergism with DT for computing key metrics and indicators related to fatigue life. Again,

one of the remaining issues identified by the authors relates to the computational needs due to very small probabilities-of-failure estimation, multi-scale modeling or SHM implementation.

Another important issue in implementing DT is data collection as it represents the entry point to update models throughout the product life cycle. Sensors play a key role in DT and the recent development of virtual sensors provide the opportunity to measure quantities unavailable with classical sensors. Navigation sensors and actuators have been an early focus in the aviation sector as replacing physical sensors would allow to reduce hardware redundancy and acquisition cost, or to be part of fault detection methodologies [8]. Thus, virtual attitude sensors [9, 10], virtual sensors of angle-of-attack [11, 12], virtual Calibrated Air Speed sensor [12] or wing angle virtual sensor [8] have been proposed for sensor failure detection and compensation. Virtual sensors are also of particular interest for health monitoring. Virtual sensors have been proposed for damage detection of an aircraft wing [13] or for prognostic and health management of aircraft tanks [14, 15]. The implementation of virtual sensors can rely on physics-based approaches [9, 10, 14], on signal processing approaches as AutoRegressive models [8, 11], on data fusion techniques [12] or on machine learning techniques [15, 13, 16]. Machine Learning techniques, or data-based techniques more generally, have received strong interest in many applications fields in recent years thanks to the widespread availability of data and the computational progress to manage these amount of data. The accurate numerical simulation of an entire aircraft in an airflow representative of the actual flight conditions is hardly conceivable due to computational resources it would require. While learning algorithms could be computationally intensive in training phase, the inference is fast in production phase making them highly competitive to be used in-line in comparison with physic-based simulations. Dynamic data-driven approaches also provide an interesting framework to estimate the dynamical behavior of a system in operation. Using high-fidelity physical models computed off-line and data assimilation techniques, these approaches have been applied to the on-line prediction of the aeroelastic responses of a joined-wing aircraft [17, 18, 19]. Nevertheless, these approaches require advanced knowledge of the studied system and experts to implement relevant physical models. Learning techniques overcome these constraints. In the context of this paper, the access to a large industrial data set of aircraft test flights naturally steers this work toward learning-based approaches to develop a virtual sensor to measure structural mechanical stress. The experimental characterization of the structural aircaft stress of all commercial aircrafts is mandatory for any manufacturer for reasons of certification. This guarantees the existence of data to generalize the proposed approach to any commercial aircraft.

In a general context of industrial application, sensors data are mainly considered as time series and two categories of problems can be identified: the prediction of futures values of the sensor's measurand using recent and seasonal values, referred to as time series forecasting problems, and the prediction (not necessarily in the future) of a the continuous value of the sensor's measurand from a set of continuous features described by univariate or multivariate time series, referred to as time series regression problems [20]. Methods used for time series forecasting range from "naive" persistence models to deep learning approaches, including regressive models and classical machine learning models (random forests, support vector machines, etc). Early in the 70s, regression algorithms have been explored, ARMA (Auto-Regressive Moving Average)[21, 22], and these models are still an approach often deployed in time series forecasting. These models aim to describe the auto-correlations in the data and captured a sequence of different temporal structures. Extension of ARMA models like ARIMA (Auto-Regressive Integrated Moving Average), SARIMA (Seasonal ARIMA) or ARIMAX (ARIMA with exogeneous features) allows to generalize this approach. Linear regression models and their penalized version, Lasso [23]

and Ridge[24], are often used as a first "baseline" in forecasting problems. Machine Learning regressors have also been widely investigated in last decades for time series forecasting and regression [25, 26]. As a biref overview, Random Forests (RF) models, a set of decision trees algorithms that can be used for classification and regression predictive modeling, are frequently used because of their simplicity of use and their good performances [27, 28, 29]. Gradient boosting, another decision tree-based supervised learning algorithm [30] and its high performances counterparts: Extreme gradient Boosting (XGgBoost) [31] and Light Gradient Boosting Machine (LightGBM) [32, 33], are also widespread for these applications. Performances of these algorithms can further be improved by the stacking of RF and boosting algorithms [34]. Artificial Neural Networks (ANN), thanks to their ability to address nonlinear processes in time series, can outperform classical statistical models on time series prediction tasks, as shown by Werbos [35]. More recently, ANN and notably the family of generative models like autoencoders, raised a strong interest in time series forecasting. Long Short Term Memory (LSTM) algorithms are built to process sequential data and are thus well suited for time series forecasting. They clearly outperform auto-regressive models such as ARIMA [36] but the computational time and memory requirements (to keep the entire sequence in memory) are a major limitation of these algorithms. The interpretability of these models is also limited [37]. Among deep learning techniques, recurrent neural networks (RNN) or one-dimensional convolutional neural networks (CNN) (usually used for image classification) have been applied to time series [38]. These methods allow to extract non-trivial pattern in data but also remain complex and difficult to configure.

This work is part of the "Paris Region Challenge AI for Industry 2020" co-organised by the french *Ile-de-France* region and the aircraft manufacturer Dassault Aviation. The aim of this challenge is to develop a virtual sensor based on learning techniques to estimate the local mechanical loads of a Falcon business jet from flight instrument only. This challenge took place over a constrained time period of almost three months at the beginning of 2021. This paper aims to present the solution proposed by the consortium consisting of Aquila Data Enabler (www.aquiladata.fr) and the *Laboratoire Quartz* from ISAE-Supméca, winner of this competition. This paper is organized as follows. After presenting the context of the challenge (section 2), the baseline proposed to address this challenge will be detailed (section 3). The numerical experiments performed during the challenge to assess the performances of the proposed approach will be presented (section 4). These results have been achieved on a reduced set of actual aircraft flights and were constrained by challenge rules (no direct access to the full dataset of aircraft flights). Section 5 concludes on the opportunities offered by learning techniques to provide virtual sensors for mechanical stress monitoring of aircraft structures.

## 2. Challenge use case presentation

Dassault Aviation is a French aerospace company specialized in the development and manufacturing of military aircraft and business jets. This challenge focuses on the business jet activity and aims at developing an original virtual sensor for the prediction of the mechanical stress in different parts of the aircraft structure. Dassault aviation has an important database of test flights established for the development and validation of their Falcon business jets. Test aircraft are equipped with conventional aircraft on-board instruments supplemented with numerous extra sensors as temperature sensors, pressure probes, pitot tubes, potentiometers or strain gauges. In particular, strain gauges, glued on different structural parts of the test aircraft, measure the mechanical stress of the aircraft in different flight phases. Such data are mandatory for certification purposes and for the validation of numerical models used in conception and design phases.

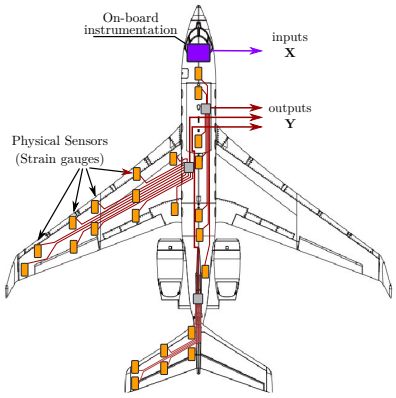



Figure 1: Schematic diagram of instrumentation layout. The exact position of gauges are unknown to the challenge participants.

Commercial aircraft are not equipped with strain gauges, only regular maintenance operations monitor the structural health of the aircraft. The main objective of the challenge is to develop a virtual stress sensor to predict the mechanical stress of commercial aircraft from the knowledge of on-board instruments only.

Defining $\mathcal{T}$ as the set of univariate time series, a time series is defined by $(\mathbf{t}, \mathbf{x}) = (t_i, y_i)_{i \in [\![0,T]\!]}$ where $T$ is the duration of the time series. The dataset provided in the challenge involves data from $n_{in} = 130$ on-board instrument signals (inputs) and $n_{out} = 67$ strain gauges spread on different parts of the aircraft (outputs). Defining $n_f$ the number of flights in the dataset, the dataset is thus composed of $(n_{in} \times n_f)$ time series $X_{jk} = (\mathbf{t}, \mathbf{x})_{j,k} \in \mathcal{T}$ and $(n_{out} \times n_f)$ time series $Y_{jk} = (\mathbf{t}, \mathbf{y})_{j,k} \in \mathcal{T}$. The dataset will be referred to as $D = \{(X_k, Y_k)\}_{k \in [\![1,n_f]\!]}$ where $X_k = \left\{X_{jk}\right\}_{j \in [\![1,n_{in}]\!]}$ and $Y_k = \left\{Y_{jk}\right\}_{j \in [\![1,n_{out}]\!]}$. The dataset will be divided in two subsets referred to as $D_{train}$ and $D_{test}$ for training and evaluation datasets. The sampling frequency of each sensor is constant over each flight but sensors are not synchronised and can have different sampling rates. Data are heterogeneous as sensors can return real, integer or boolean values. No other information on the acquisition system are available, the input signals are considered as « well conditioned and pre-processed » by the aircraft manufacturer. Inputs data (on-board instrumentation) are categorized in 10 groups corresponding to Inertial Reference System (IRS - 3 redundant systems, 14 sensors each), Air Data Systems (ADS - 4 redundant systems, 7 sensors each), masses (8 sensors), engines (6 sensors), cabin (1 sensor), landing gear (7 sensors), radio altimeter (1 sensor), auto-pilot (3 sensors), primary and secondary control surfaces (resp. 13 sensors and 17 sensors) and outputs data (gauges) are categorised in 5 groups corresponding to 4 localizations on the aircraft (16 gauges on the fuselage, 14 on wings, 19 on pitch, 3 on fin) and to actuators (15 gauges). A schematic diagram is presented Figure 1 to illustrate the instrumentation layout. While gauges signals were labeled according to their position on the aircraft structure and to the type of loads being measured, their exact position were unknown to the challenge participants.

The objective of the challenge is thus to find a (learning) model $f : X \to Y$ minimizing the

performance function

$$P(f, D, \mathrm{L}) = \frac{1}{|D|} \sum_{(X,Y)\in D} \mathrm{L}(f(X), Y), \tag{1}$$

where $\mathrm{L} : \mathcal{T} \times \mathcal{T} \rightarrow \mathbb{R}$ is the evaluation metric. The learning problem is thus a time series regression task.

The metric considered in the challenge to evaluate the quality of the model proposed by each candidate is the Mean Absolute Error (MAE) defined by

$$\mathrm{MAE} = \sum_{k}^{n_{val}} \frac{|(f(X_k) - Y_k)|}{n_{val}}, \tag{2}$$

where $f(X_i)$ are the predicted outputs obtained by model proposed by the candidate and trained on $D_{train}$ and $n_{val}$ is the number of flight chosen to be part of the validation set $D_{val}$.

The whole dataset was hosted on Université Paris-Saclay's server and was available through the *CodaLab* plateform of Chalearn www.chalearn.org [39]. $D_{train}$ includes a total of 117 readable flights and $D_{val}$ is made of a set of sub-sequences extracted from 186 flights. Flight in $D_{val}$ are different than flights in $D_{train}$. The full data set represents 70 GB. Each participant was able to use one compute worker on the *Codalab* platform which includes 1 GPU NVIDIA RTX 2080Ti, 4 vCPUs and 16 GB DDR4 RAM. It is worth noting that the whole dataset was not directly available to the candidates for confidentiality reasons. A reduced set of three flights, referred to as *Starting Kit* in the remainder, was however available for data exploration and algorithm prototyping.

## 3. Proposed baseline

Data exploration is a fundamental step in Data Science projects. In order to propose a baseline adapted to the studied problem, data have to be understood and consolidated. This section summarizes the main steps of the approach used during the challenge to consolidate data. After different steps of data exploration, feature engineering operations and the model architecture are defined according to data exploration conclusions. A strategy to manage sensors failure is also defined. The three flights of the challenge *Starting Kit* are considered in this section.

### *3.1. Data exploration*

Data exploration has been performed using a simple and pragmatic approach based on the visualisation of the time signals associated to each available sensors of the *Starting Kit*. The simultaneous visualization of sensors signals grouped by redundancy or by the nature of the physical quantity measured by the sensors enables the identification of significant anomalies and faults. Disabled inputs or outputs have thus been identified on some flights of the *Starting Kit*. An anomaly in the behaviour of 4 gauges has been detected by noticing a significant discrepancy in signals of those sensors (sensors measuring strain on the same substructure with same loads are expected to be correlated). It has been confirmed by challenge organizers that flights are ordered in time in the full dataset hosted on the challenge platform and the identified sensors failed after a certain flights. As a consequence, the strategy chosen to manage effectively the training data consisted in dropping entire flights data when output sensors are disabled. A pragmatic trial and error procedure identified the number of flights with missing gauges data. The full train set includes a total of 117 readable flights, slightly more than half of the flights contain all 67 gauges

columns that need to be predicted, meanwhile, the remaining flights lack at least one of a set of four gauges. To maximize the use of available data, a modelling strategy with one model per output has been adopted and each model is trained on a training set made of flights where the gauge of interest is available. Thus, the models of 4 gauges were trained on half of the available flights while the models of the remaining gauges were trained with all available flights. This basic strategy is justified by the constraints of the challenge and the restriction on the access to the training set (blind training of the model) and would be improved with a full access to the training data set. A review of more sophisticated techniques dedicated to anomaly detection in sensor systems has been proposed by Erhan et al. [40]. For disabled inputs, redundancies in flight instrumentation allowed to directly drop features without any loss of information.

Data exploration also helps to appreciate the desynchronization between raw times series provided by sensors as well as the variability of the sampling rate of all the aircraft sensors. It highlights the need of a prepocessing step on raw data. An interpolation and a resampling of raw data have thus been considered. Three classical interpolation scheme (back-fill, front-fill and nearest-fill) have been implemented and tested. Data from the *Starting Kit* and a LightGBM model where considered for this study. No significant improvement of the model performances in terms of overall MAE have been noticed, a back-fill interpolation method has thus been selected to fill in missing data. Data resampling on interpolated data is then straightforward. The complete dataset available on the challenge plateform *CodaLab* involves close to 70GB and cannot fit in the memory (RAM) of the platform's servers (16 GB RAM available). Two strategies could then be considered to train a model on the available data. The first is to train the model by batch, the second is to down-sample the data to fit the memory limitation of the computation resources. Models considered by the consortium during the challenge were not implemented with batch training option (in the version available on official repository at the time of the challenge). A consistent comparison of the models MAE performances in model selection phase is obtained by down-sampling, this strategy have thus been chosen. Random sampling and uniform sampling strategies have been tested. For the first, strategy data are randomly sampled over each flight from uniform distribution. For the second strategy, flight are uniformly sampled at a (low) fixed frequency. A specificity of the data considered in this challenge is that sensors data are acquired on test flights with a wide range of varying mission profiles. As a consequence, each flight of the dataset has a variety of flight phases with unknown duration and order. A uniform sampling strategy with low sampling frequency is thus selected to ensure a relevant representativeness of every flight phases. Different sampling rate have been tested on the *Starting kit* dataset with different tree-based models. A sampling rate of 0.5Hz has been selected as this rate did not decrease significantly the overall MAE obtained on the *Starting Kit* dataset and fitted with memory constraints of the *Codalab* platform's server. In order to improve the prediction of the models on extreme events such as gust of wind or turbulence, characterized by short term phenomena with higher dynamics, a non uniform sampling strategy may be considered in order to densify the number of samples related to these phenomena. This strategy has not been implemented due to the challenge constraints (full data set not directly available to define a relevant strategy for short term phenomena) but will be addressed in further work.

### *3.2. Data preparation*

Redundancy of critical on-board instruments are mandatory to satisfy safety requirements of civil aviation. Nevertheless, this redundancy is not meaningful according to the learning task of this challenge. Groups of sensors have thus been defined based on features redundancy, for instance, sensors of Air Data System (ADS) or sensors of Inertial Reference System (IRS), have

been grouped by type. Signals of redundant sensors have been merged, reducing the number of inputs with the aim to increase learning efficiency.

Correlations in data have also been studied to make the best possible use of available data. Correlations between inputs identify correlated inputs and potential redundancy of information between features. Uncorrelated inputs are thus sought and a strategy could be defined to merge correlated data into meta-features to improve training performances. Correlation between outputs could also be used to improve the modelling strategy. Independent models could be trained when outputs are not correlated or relations between models outputs could be considered to improve model regularization and decrease sensitivity of the model to abnormal data (outliers or extreme events). Finally, correlations between inputs and outputs could also be considered to identify inputs mainly contributing to each output and, inversely, to discard inputs with no direct contribution to outputs. Input dimension considered for each model could therefore be adapted to improve model performances.

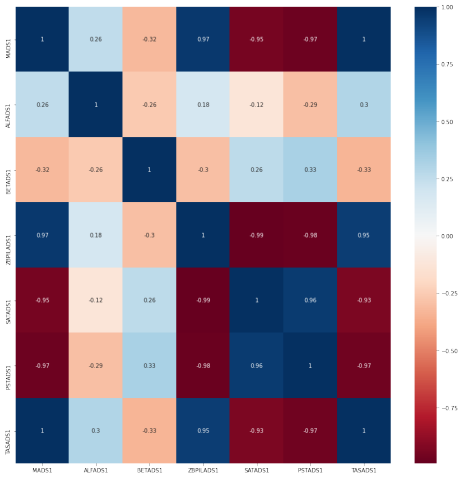



Figure 2: Air Data System (ADS) sensors correlation matrix (computed with data from a single flight of the *starting kit*)

Correlated inputs have been identified. For instance, correlation between sensors from Air Data System (ADS) can be noticed on the correlation matix (Pearson's correlation) presented Figure 2. Mach sensor, pressure-altitude sensor, true air speed sensor, static air temperature sensor and static air pressure sensor are thus strongly correlated (absolute correlation coefficient greater than 0.95). These correlation can be related to air properties and notably the standard atmosphere model.

Correlation between outputs (gauges) have been investigated. For example, fuselage gauges correlation matrix is presented Figure 3. Three groups of correlated gauges can be defined. These correlations are in accordance with the description of the gauge type of load and their position on the fuselage. An exception is however noticed for the gauge labelled *FFUS4*. This gauge is known to be dedicated to flexion measurement and is expected to be correlated with *FFUS1*, *FFUS2* and *FFUS3* (at least) due to its position on the fuselage. *FFUS4*'s low correlation score is in contradiction with these assumption. This could be explained by a fault in the sensor or a mistake in the gauge description (position and load). It points out that the definition of the model architecture or the data preprocessing based on this correlation study should be made carefully.

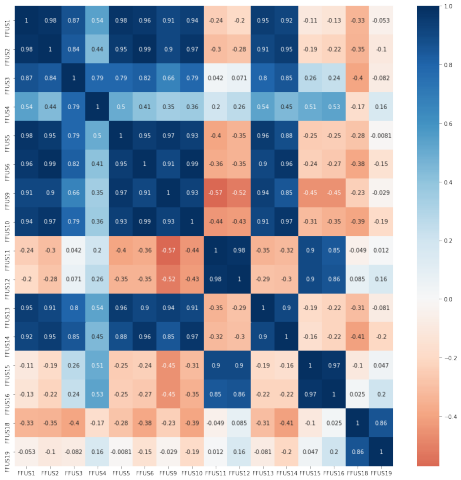


Figure 3: Fuselage gauges correlation matrix (computed with data from a single flight of the *starting kit*)

To take into account the correlations between outputs, a regularization layer of regressors can be introduced to automatically learn relation between gauges outputs. This aspect will be detailed in model definition section 3.3.2.

Finally, a correlation study has been conducted between outputs and inputs groups. This analysis identified different reduced sets of input features correlated to the different outputs groups. These observations could have been used to reduce the set of inputs considered to train each model of gauge. Nevertheless, this correlation study was conducted on data from the *starting kit* and should have been tested and validated on the entire data set to consider reduced sets of inputs adapted to train each model of gauges.

Correlation between data could be interpreted according to aircraft physics in its environment and the observed correlation should be integrated in the modelling strategy only if physics based interpretation are fully relevant (the opposite would risk to miss meaningful data). Due to challenge constraints, correlation study was not possible on the full dataset. The exact configuration of the sensors considered in this study, notably the position of the gauges on the aircraft structure, were also not available. Due to high uncertainties on the expected behaviour of sensors signals, physics based interpretations were therefore limited and not pursued further in this work.

### *3.3. Feature engineering and Model*

This section focuses on feature engineering operation and model selection inferred from data exploration presented section 3.2

#### *3.3.1. Phase identification: Clustering*

The main assumption of the proposed approach is that aircraft in-flight physics, particularly aircraft structural loads, varies little for a given phase of flight. According to this assumption, an efficient identification of the operating phases would enable to reduce dramatically the complexity of the learning task. This automatic phases of flight identification also aimed to improve model explainability. The physical behavior of the aircraft depends on many parameters, nevertheless the ADS and IRS sensor groups contain a coarse yet sufficiently complete image of the

aircraft physical state to provide a pertinent clustering. For each of these two groups of sensors a clustering model based on a $k$-means algorithm [41] has been trained.

Given a training set $\mathcal{S} \subset \mathbb{R}^n$ of samples, the $k$-means algorithm aims to find a set of $k$ centroids $\mathcal{M} = \{\mathbf{c}_1, \ldots, \mathbf{c}_k\} \subset \mathbb{R}^n$ minimizing the cost $C$:

$$C = \sum_{\mathbf{c} \in \mathcal{M}} \sum_{\mathbf{x}_i \in \mathcal{S}_i} \|\mathbf{x}_i - \mathbf{c}\|^2,$$

where $\|\mathbf{x}\|$ denotes the L2-norm of the vector $\mathbf{x}$ and $\mathcal{S}_i$ are partition of $\mathcal{S}$. The learning algorithm of the $k$-means is well-known and fully described in the literature and will not be detailed further.

This unsupervised learning technique has been used to create two additional features (the cluster label) added as input of the gauges models. Denoting $\mathcal{X}_{ADS}$ the set defined by the concatenation over all flights of the multi-variate time series associated with sensors of ADS and $\mathcal{X}_{IRS}$ for sensors of IRS, one clustering algorithm is trained on $\mathcal{X}_{ADS}$ and another on $\mathcal{X}_{IRS}$. ADS and IRS respectively involve 7 and 14 sensors each and merged signals (according to section 3.2) are considered. The dimensions of $\mathcal{X}_{ADS}$ and $\mathcal{X}_{IRS}$ are $(T_f \times 7)$ and $(T_f \times 14)$ respectively, where $T_f$ is the cumulative duration of all flights. The clustering of ADS and IRS sensors are made using instantenous data. To determine the number of clusters $k$ for ADS and IRS clustering, a hierarchical clustering technique has been trained on the *starting kit* data and a dendrogram of the clusters hierarchy has been plotted to identify a suitable number of clusters for the k-means algorithm. 16 clusters were used in the final pipeline.

### *3.3.2. Model*

The gauges values are predicted using an ensemble model which consists of two regressors stacked together. The first regressors takes instantaneous values of the pre-processed sensor data, referred to as $\mathbf{x}$ on Fig 4, as inputs and their outputs are the gauges values, referred to as $\hat{\mathbf{y}}$ on Fig 4. The models are trained by minimizing the loss function between $\hat{\mathbf{y}}$ and the reference values of each gauges $\mathbf{y}^{ref}$. The second regressors take the prediction of the first regressors $\hat{\mathbf{y}}$ as inputs and the models are trained by minimizing the loss function between $\tilde{\mathbf{y}}$ and the reference values of each gauges $\mathbf{y}^{ref}$. Their predictions are the final prediction of the model. The detailed model is presented in Figure 4.

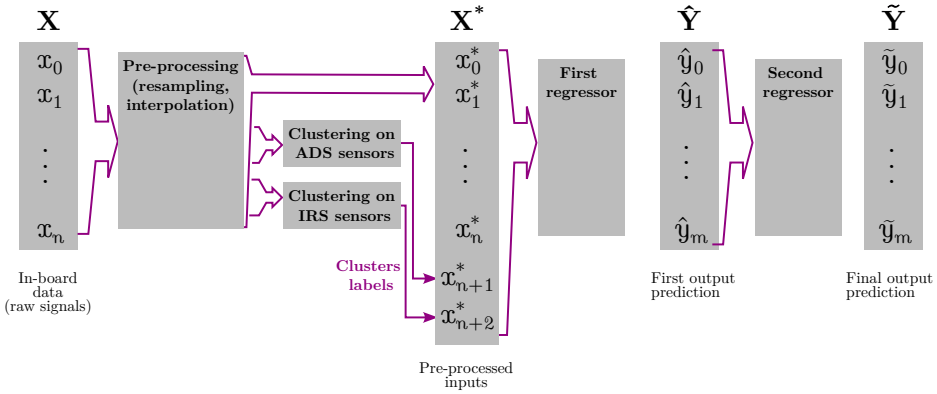



Figure 4: Flow chart of the proposed pipeline with stacking strategy (**x** vector of instantaneous values of the pre-processed sensor data, $\hat{\mathbf{y}}$ vector of instantaneous gauge values predicted by the first layer of regressors, $\tilde{\mathbf{y}}$ vector of instantaneous gauge values predicted by the second layer of regressors.

For the basic regressors, a light gradient boosting machine (LightGBM or L-GBM in the remaining) model is considered. L-GBM is an advanced Gradient Boosting Decision Tree (GBDT) model developed by Microsoft in 2017 [42]. The main idea of this algorithm is to accumulate a series of weak regressors $h(x)$ into a strong regressor $f(x)$ through forward addition (3).

$$f(x) = \sum_{m=1}^{M} \beta_m h(x; \Theta_m) \tag{3}$$

Where $x$ is an input data sample, $h(x; \Theta_m)$ represents the $m^{th}$ weak regressor with parameter $\Theta_m$. $\beta_m$ weight each weak regressor to compute the prediction of the global model. Compared with other gradient boosting decision tree models, LightGBM is optimized in terms of memory and computing efficiency.

The second layer of regressors considered in the stacking approach is a linear ridge model. This regressor has been chosen for its simplicity and its robustness.

## 4. Experiments

This section introduces model performances on *Starting Kit* data and on training and validation sets provided on *codalab* platform. It also present the investigations performed during the challenge to evaluate the impact of each elements of the proposed baseline on the gauges prediction performances. Notably, the impact of the clustering step, of the sampling rate and of the model used for the first layer of the staking model are presented in this section.

### *4.1. Overall performances of the proposed pipeline*

#### *4.1.1. Performances on Starting Kit and model pre-selection*

In order to choose the best predictive model, a first evaluation of different algorithms has been carried out on the *starting kit* by using data from flights 1 and 2 for training and flight 3 for testing. Results are shown in figure 5. The metric considered to evaluate models prediction performances is the MAE. The following models have been tested, their hyperparameters are indicated in parentheses:

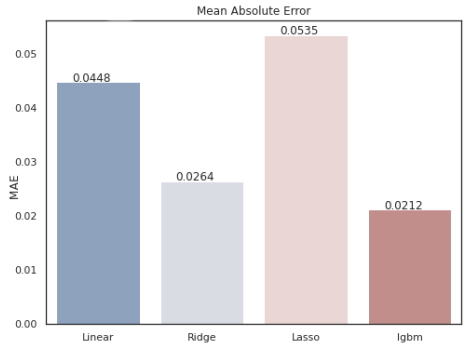



Figure 5: Comparison of Mean Absolute Error (MAE) values of *Linear*, *Ridge*, *Lasso* and *light gradient boosting machine (lgbm)* regressors trained and tested on *starting kit* data.

- regression models: Linear, Ridge (alpha = 20) and Least Absolute Shrinkage and Selection Operator, LASSO (alpha = 1)
- tree based model: LightGBM (learning rate=0.25, traditional Gradient Boosting Decision Tree, Maximum tree leaves for base learners = 31, Number of boosted trees to fit = 100)

LightGBM model gives the best prediction results on *starting kit* data with a Mean Absolute Error value of 0.0212 compared to other algorithms.

*4.1.2. Performances on full dataset (Codalab)*

The performances of the proposed pipeline have also been evaluated after training on the whole training set of 117 flights available on *CodaLab* platform. The accuracy of the prediction is evaluated with the MAE metric on a validation set of sequences from 186 flights. As only in-flight data were evaluated in the framework of the challenge, ground phases have been dropped for model training.

The main advantages of the algorithms considered in the proposed pipeline are their training and prediction low complexities. Thus the computational time for these two tasks are low as illustrated in Table 1.

The accuracy of the prediction of the proposed pipeline on validation and testing sets are also very interesting. With an overall MAE on validation and testing sets of respectively 0.0152 and 0.0140 as presented in table 1, the proposed model was on the top 3 of the most accurate models on the challenge leader board during the prototyping phase. In order to better assess the MAE score of the proposed approach, it can be noted that data provided by Dassault Aviation were normalized (to ensure the confidentiality), all outputs (gauge signals) considered to train the model were thus in the range of 0 to 1.

The prediction performances have also been evaluated depending on the gauges location on the aircraft and for specific flight phases associated with heavy loads on the aircraft. Even though the average MAE remains low in all cases, it can be noticed in Table 2 that the model performance is lower for gauges on fuselage and pitch than for gauges on wings, fin or actuators. Concerning flight phases, the average MAE remains lower than 0.015 at any flight phase. A first study presented the design and analysis of the data science competition. This study compared the performances of the different models proposed by the ten teams which took part in the challenge [43]. Participating teams involved could be made up of industrials and academics members and

| | MAE | Learning duration (s) | Testing Duration (s) |
|---|---|---|---|
| Validation | 0.01519 | 1280 | 6635 |
| Test | 0.01402 | 1280 | 5750 |

Table 1: Mean Absolute Error (MAE) and time of the proposed pipeline on validation and testing sets.

were selected prior to the challenge on the basis of their skills in data science and machine learning. This should ensure the use of state-of-the-art algorithms. The pipeline proposed in this work was ranked in the top three results. The study concludes that the performances of the participants ended up very close to one another by the end of the challenge.

| Gauges location | MAE | Flight phases | MAE |
|---|---|---|---|
| Fuselage | 0.02050 | Maneuver | 0.01405 |
| Wings | 0.01014 | Gust | 0.01505 |
| Pitch | 0.01467 | Turbulence | 0.01229 |
| Fin | 0.01023 | | |
| Actuator | 0.01155 | | |

Table 2: Average Mean Absolute Error (MAE) obtained on the validation set for different gauge location and different flight phases.

### *4.2. Experimental investigation for model improvement*

This section is dedicated to the experimental tests performed during the challenge to improve the model performances. We want to note here that the challenge constraints where such that it was only possible to train a model on the train set (hosted on *Codalab* platform) and to evaluate the MAE on the validation set (also hosted on *Codalab*). Challenge rules prohibited the access to detailed logs on the data or on the model, no information on the model weights or on the prediction over flights were available. For this reason, a trial-error strategy with global MAE on validation set as sole indicator has been used to investigate the impact of the different elements of the proposed baseline. Tests were performed on the *starting kit* to speed up the developments and guide the investigations on the validation set, nevertheless, their generalization to the validation set has not been systematic due to time constraints of the challenge agenda.

#### *4.2.1. Impact of clustering*

In this section we discuss the impact of the clustering on the proposed pipeline. The inputs of the clustering step are features of the ADS and IRS groups (clusters have been computed on both groups separately, see section 3.3.1). The use of clustering as input features improve the MAE on both the *starting kit* and the validation kit for various algorithms. The impact of the clustering step on two basic regression algorithms, Adaboost and Ridge algorithms, has been evaluated on the validation set (*CodaLab*). Table 3 presents the improvement on the MAE and thus on the prediction accuracy of these two algorithms.

Training the Adaboost and Ridge algorithms with inputs augmented with ADS and IRS cluster's labels increase the overall MAE on the validation set. Due to challenge's agenda, the final pipeline has not been tested without the clustering on the validation set. It is thus not possible to quantify the impact of the clustering on the final pipeline MAE when the model is trained on the full diversity of flights. Nevertheless, the final pipeline was tested with and without clustering on

| | MAE without clustering | MAE with clustering |
|---|---|---|
| Adaboost | 0.033 | 0.026 |
| Ridge | 0.028 | 0.023 |

Table 3: Mean Absolute Error (MAE) for two regression algorithms (Ridge linear regression and Adaboost regression) with or without clustering when learned on the training set and tested on the validation set.

the *starting kit*. Its benefits in terms of MAE were less meaningful as illustrated in Table 4. The number and diversity of flights phases in the *starting kit* is much smaller than in the validation set, it is thus not possible to conclude on the benefit of the clustering on the MAE score of the final pipeline when trained and evaluated on the *Codalab* platform. Further investigation would be necessary. Notably, the analysis of the model weights after training on the full training set would help to conclude on the influence of the clusters features on the regressors training. During the challenge, no access was given to the model weights. This would be the object of further works on an equivalent dataset.

| | MAE without clustering | MAE with clustering |
|---|---|---|
| Final Pipeline | 0.0287 | 0.0285 |

Table 4: Mean Absolute Error (MAE) for our final pipeline with or without clustering when learned and tested on the *starting kit*.

While the improvement of the prediction accuracy induced by the clustering cannot be quantified, it is highlighted that the clusters labels add explainability to the proposed model. The clustering step aims to identify patterns in ADS and IRS signals associated with phases of flight. Without going so far as identifying the underlying physics of each phases of flight, this step automatically labels the phases of flight with the aim of specializing the regressors. Clusters labels also facilitate the analysis of the model using conditional extraction of the model predictions. .

*4.2.2. Impact of core algorithms for regression*

The strategy adopted for model selection was supported by a permanent focus on submissions performances. An analysis of the results obtained for each submission on *CodaLab* platform has been carried out in order to improve and refine the proposed model.

The scores (MAE) and run time of different models trained using the complete training set and tested using the esting set available on *CodaLab* have been investigated. The L-GBM model has provided the best prediction results with a Mean Absolute Error value of 0.0203 compared with *Random Forest*, *Ridge*, *Multi-Layer Perceptron* and *Adaboost* algorithms. L-GBM also had the second lowest training duration compared to other models as illustrated in Figure 6. The hyperparameters considered for each algorithm are 100 trees with maximum depth of 20 for Random Forest, regularization factor of 20 for Ridge, two hidden layers with respectively 50 and 30 neurons and rectified linear unit function (relu) for activation function for MLP, Adaboost use decision tree regressor as base estimator and a learning rate of 0.5.

An extensive analysis of the second regressor would have ensured the best choice of algorithm for the second regressor but at the expense of a time consuming submission on the Codalab

platform. The improvement of the model prediction in terms of MAE related to the stacking strategy has not been quantified on the full dataset during the challenge. By comparing the MAE of the LightGBM algorithm trained and tested on Codalab platform (MAE LightGBM = 0,0203) and the MAE of the final pipeline (MAE final pipeline = 0.01519), it is possible to infer the benefit of the model stacking although it is not possible to quantify it precisely.

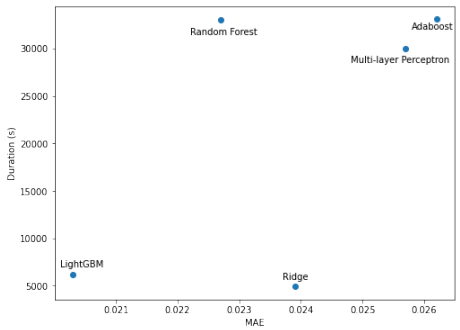



Figure 6: Mean Absolute Error (MAE) values and run time of *Ridge*, *Random Forest*, *Mulit-Layer Perceptron*, *Adaboost* and *light gradient boosting machine (LightGBM)* regressors trained and tested on *CodaLab* platform.

### *4.3. Explainability of the results*

#### *4.3.1. Explainability with SHAP*

Decision tree-based algorithms conserve a natural explainability. The SHAP library [44] provides tools for machine learning models interpretability. It allows to easily compute importance feature of regression or tree-based models based on Shapley values, a widely used approach from cooperative game theory. SHAP thus compute the Shapley value of each input with the aim to dentify the most informative relationships between the input features and the predicted outcome (the stronger the absolute value of the Shapley value is, the stronger the feature contribute to the outcome). A positive Shapley value shows that the associated feature value contribute positively to the regression output value whereas a negative Shapley value indicates that the associated feature value tends to decrease the regression value. The Shapley values can be used to illustrate the contribution of each feature to the gauges models output. In the context of this paper, this library is used to evaluate the most contributing input to the prediction of a single gauge model. This analysis has been applied to several gauges models during the challenge but will not be detailed in this paper. Figure 7 presents the time series of a gauge glued on the aircraft fuselage and the prediction provided by the proposed pipeline after training. The model has been trained with the *starting kit* and the flight 3 is considered for the prediction. The SHAP library allows to identify the input sensors contributing the most to the output of the model. The normalized gauge values are plotted in Figure 7i and the main impacting sensors according to the Shapley values are presented in Figure 7ii and Figure 7iv in order to illustrate the analysis.

The shapley values have been computed for two distinct points of the flight. The first (blue dotted line) corresponds to the beginning of the cruise phase, the second (red dotted line) corresponds to an approach phase. Figures 7iii and 7v present the SHAP values for these flight points. It can be noticed that the gauge prediction is strongly impacted by the speed of the aircraft (MACH), the pressure altitude (PALT) and the static air temperature (TSTA). The Angle-of-Attack (ALPHA) also have a significant contribution but in cruise phase only. It should be

mentioned that this section aims to illustrate the compatibility of the proposed pipeline with that kind of analysis tools and the advantage this would represent for the exploitation of the model in production by the aircraft manufacturer. The SHAP analysis has been performed on the model trained on the *Strating Kit*, the observations presented here must be considered with regards to the performances of the model trained on the *Starting Kit* and should not be considered as generic results, the model must be trained on the full dataset before.

*4.3.2. Physical interpretation*

The physical interpretation of the results presented in section 4.3.1 is difficult due to the complexity of the aircraft loading and the number of parameters impacting the load. The lack of knowledge on the exact position of the gauge is also a limitation to allow rigourous physical interpretations. In a first line of thought, this section gives insights on interpretations of the results provided by the shapley values analysis (sec. 4.3.1). The fuselage gauge considered in the previous analysis aims to measure strain induced by the fuselage bending. The lift and drag of the aircraft are directly linked to the aircraft speed (MACH) and the angle-of-attack (ALPHA) and these loads tends to bend the fuselage. The dynamic pressure $P_d$ is also one of the key parameter of the aerodynamical loads. It directly impacts the bending loads of the aircraft fuselage. $P_d$ is defined by:

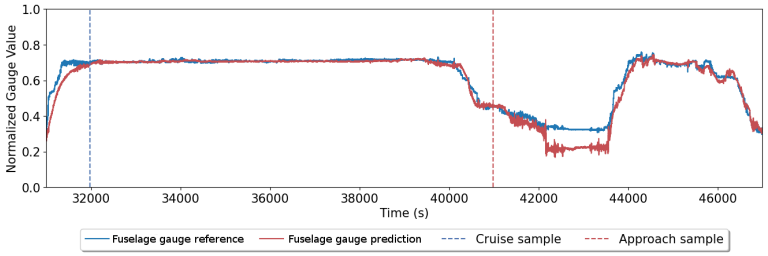
$$P_d = \frac{1}{2}\rho V^2, \tag{4}$$

where $\rho$ is the fluid mass density, and $V$ is the flow speed of the aircraft. $V$ is linked to the aircraft speed (MACH) and $\rho$ is linked to the pressure altitude (PALT) and the static air temperature (TSTA). This could explain, in a first approach, the link between the model and the aircraft physics highlighted by the shapley value analysis.

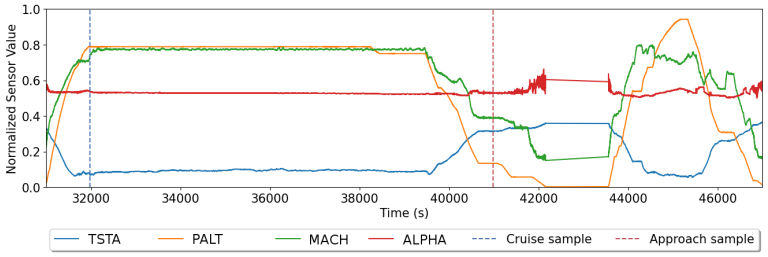
## 5. Conclusion and outlook

The solution developed to address the challenge proposed by Dassault Aviation and the french *Ile-de-France* region has demonstrated the feasibility of the concept of virtual sensor for the prediction of the structural mechanical stress of an aircraft from the measurement of flight instruments available on any commercial aircraft. The solution proposed in this work relies on the combination of a clustering algorithm to identify flight phases and a two stages regressors based on light gradient boosting machine and Ridge algorithms. The performances of the proposed approach in terms of training frugality and prediction accuracy have been demonstrated on a training set of 117 flights and a test set of flights sequences from 186 flights. During the prototyping phase of the challenge, the proposed model reaches a MAE of 0.01520 on the test set. During the validation phase of the challenge, only a single code submission was allowed for each team, sequences used in test set for this phase differed from sequences provided during the prototyping phase and the proposed model obtained a MAE of 0.0140. The interpretability of the trained model is another advantage of the proposed approach as it will enable a better understanding of the actual loads applied to the aircraft when using the virtual sensor in service phase. With the prospect of using the proposed virtual sensor on actual commercial aircrafts, the fast inference of the trained model allows to consider in-line computation to predict the structural stress of the aircraft.

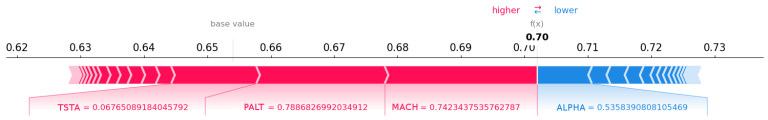
In order to exceed the performance of the solution developed during the challenge (in only two months and with no direct access to the full dataset), state-of-the-art deep learning architectures are currently under investigation by the consortium. In particular, the Variational Auto



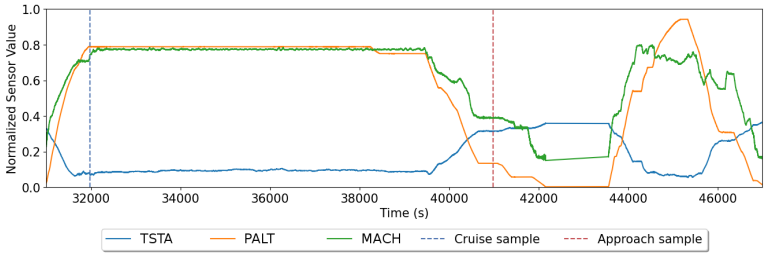

i Evolution of the considered fuselage gauge on a flight from the *starting kit*.



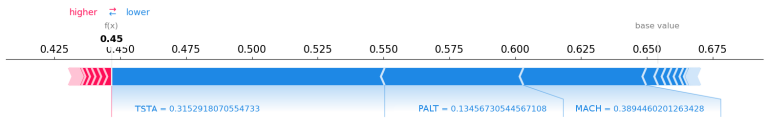

ii Evolution on a flight from the *starting kit* of the sensors that contribute most to the prediction of the considered fuselage gauge in cruise phase




iii SHAP values of the sensors that contribute most to the prediction of the considered fuselage gauge during cruise phase. The time of the considered sample is defined on fig. 7i




iv Evolution on a flight from the *starting kit* of the sensors that contribute most to the prediction of the considered fuselage gauge in approach phase




v SHAP values of the sensors that contribute most to the prediction of the considered fuselage gauge during approach phase. The time of the considered sample is defined on fig. 7i

Figure 7: Comparative study of the features impacting the prediction of considered fuselage gauge during cruise phase and approach phase.

Encoder (VAE) [45] architecture is studied. The possibility to explicitly structure the VAE latent space to give it desirable properties appears as a good opportunity to hybridise VAEs with physic-based models. This approach aims at taking advantage of the capabilities of deep-learning techniques while providing interpretability to the model thanks to its physics awareness.

## References


[1] M. Grieves, Digital twin: manufacturing excellence through virtual factory replication, White paper 1 (2014) 1–7.

[2] B. R. Barricelli, E. Casiraghi, D. Fogli, A survey on digital twin: Definitions, characteristics, applications, and design implications, IEEE Access 7 (2019) 167653–167671.

[3] M. Liu, S. Fang, H. Dong, C. Xu, Review of digital twin about concepts, technologies, and industrial applications, Journal of Manufacturing Systems 58 (2021) 346–361, digital Twin towards Smart Manufacturing and Industry 4.0. doi:10.1016/j.jmsy.2020.06.017.

[4] E. J. Tuegel, A. R. Ingraffea, T. G. Eason, S. M. Spottswood, Reengineering aircraft structural life prediction using a digital twin, International Journal of Aerospace Engineering 2011.

[5] E. Tuegel, The airframe digital twin: some challenges to realization, in: 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference 20th AIAA/ASME/AHS Adaptive Structures Conference 14th AIAA, 2012, p. 1812.

[6] C. Li, S. Mahadevan, Y. Ling, L. Wang, S. Choze, A dynamic bayesian network approach for digital twin, in: 19th AIAA Non-Deterministic Approaches Conference, 2017, p. 1566.

[7] H. Millwater, J. Ocampo, N. Crosby, Probabilistic methods for risk assessment of airframe digital twin structures, Engineering Fracture Mechanics 221 (2019) 106674.

[8] G. Heredia, A. Ollero, Virtual sensor for failure detection, identification and recovery in the transition phase of a morphing aircraft, Sensors 10 (3) (2010) 2188–2201.

[9] A. Tomczyk, Simple virtual attitude sensors for general aviation aircraft, Aircraft Engineering and Aerospace Technology.

[10] M. T. Burston, R. Sabatini, R. Clothier, A. Gardi, S. Ramasamy, Reverse engineering of a fixed wing unmanned aircraft 6-dof model for navigation and guidance applications, in: Applied Mechanics and Materials, Vol. 629, Trans Tech Publ, 2014, pp. 164–169.

[11] P. A. Samara, J. S. Sakellariou, G. N. Fouskitakis, J. D. Hios, S. D. Fassois, Aircraft virtual sensor design via a time-dependent functional pooling narx methodology, Aerospace Science and Technology 29 (1) (2013) 114–124.

[12] G. Alcalay, C. Seren, G. Hardier, M. Delporte, P. Goupil, Development of virtual sensors to estimate critical aircraft flight parameters, IFAC-PapersOnLine 50 (1) (2017) 14174–14179.

[13] H. Salehi, S. Das, S. Chakrabartty, S. Biswas, R. Burgueño, A machine-learning approach for damage detection in aircraft structures using self-powered sensor data, in: J. P. Lynch (Ed.), Sensors and Smart Structures Technologies for Civil, Mechanical, and Aerospace Systems 2017, Vol. 10168, International Society for Optics and Photonics, SPIE, 2017, pp. 234 – 246. doi:10.1117/12.2260118.

[14] T. Shen, F. Wan, W. Cui, B. Song, Application of prognostic and health management technology on aircraft fuel system, in: 2010 Prognostics and System Health Management Conference, IEEE, 2010, pp. 1–7.

[15] A. N. Srivastava, Greener aviation with virtual sensors: a case study, Data Mining and Knowledge Discovery 24 (2) (2012) 443–471.

[16] M. Oosterom, R. Babuska, Virtual sensor for the angle-of-attack signal in small commercial aircraft, in: 2006 IEEE International Conference on Fuzzy Systems, IEEE, 2006, pp. 1396–1403.

[17] R. Kania, A. Kebbie-Anthony, X. Zhao, S. Azarm, B. Balachandran, Dynamic Data-Driven Approach for Unmanned Aircraft Systems and Aeroelastic Response Analysis, Springer International Publishing, Cham, 2018, pp. 193–211. doi:10.1007/978-3-319-95504-9_10.

[18] X. Zhao, A. Kebbie-Anthony, S. Azarm, B. Balachandran, Dynamic data-driven multi-step-ahead prediction with simulation data and sensor measurements, AIAA Journal 57 (6) (2019) 2270–2279. doi:10.2514/1.J057913.

[19] X. Zhao, S. Azarm, B. Balachandran, Online Data-Driven Prediction of Spatio-Temporal System Behavior Using High-Fidelity Simulations and Sparse Sensor Measurements, Journal of Mechanical Design 143 (2), 021701. doi:10.1115/1.4047690.

[20] C. W. Tan, C. Bergmeir, F. Petitjean, G. I. Webb, Time series extrinsic regression, Data Mining and Knowledge Discovery (2021) 1–29doi:10.1007/s10618-021-00745-9.

[21] E. George, G. M. Jenkins, G. C. Reinsel, Time series analysis: forecasting and control, Wiley, 1970.

[22] R. F. Carlson, A. MacCormick, D. G. Watts, Application of linear random models to four annual streamflow series, Water Resources Research 6 (4) (1970) 1070–1078.

[23] R. Tibshirani, Regression shrinkage and selection via the lasso, Journal of the Royal Statistical Society: Series B (Methodological) 58 (1) (1996) 267–288.
[24] A. E. Hoerl, R. W. Kennard, Ridge regression: applications to nonorthogonal problems, Technometrics 12 (1) (1970) 69–82.
[25] G. Bontempi, S. B. Taieb, Y.-A. Le Borgne, Machine learning strategies for time series forecasting, in: European business intelligence summer school, Springer, 2012, pp. 62–77.
[26] W.-C. Wang, K.-W. Chau, C.-T. Cheng, L. Qiu, A comparison of performance of several artificial intelligence methods for forecasting monthly discharge time series, Journal of hydrology 374 (3-4) (2009) 294–306.
[27] M. J. Kane, N. Price, M. Scotch, P. Rabinowitz, Comparison of arima and random forest time series models for prediction of avian influenza h5n1 outbreaks, BMC bioinformatics 15 (1) (2014) 1–9.
[28] R. Genuer, J.-M. Poggi, C. Tuleau-Malot, Variable selection using random forests, Pattern recognition letters 31 (14) (2010) 2225–2236.
[29] T. M. Oshiro, P. S. Perez, J. A. Baranauskas, How many trees in a random forest?, in: International workshop on machine learning and data mining in pattern recognition, Springer, 2012, pp. 154–168.
[30] J. H. Friedman, Greedy function approximation: a gradient boosting machine, Annals of statistics (2001) 1189–1232.
[31] J. Friedman, T. Hastie, R. Tibshirani, Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors), The annals of statistics 28 (2) (2000) 337–407.
[32] E. Al Daoud, Comparison between xgboost, lightgbm and catboost using a home credit dataset, International Journal of Computer and Information Engineering 13 (1) (2019) 6–10.
[33] C. Bentéjac, A. Csörgő, G. Martínez-Muñoz, A comparative analysis of gradient boosting algorithms, Artificial Intelligence Review 54 (3) (2021) 1937–1967.
[34] M. Massaoudi, S. S. Refaat, I. Chihi, M. Trabelsi, F. S. Oueslati, H. Abu-Rub, A novel stacked generalization ensemble-based hybrid lgbm-xgb-mlp model for short-term load forecasting, Energy 214 (2021) 118874.
[35] P. J. Werbos, Generalization of backpropagation with application to a recurrent gas market model, Neural networks 1 (4) (1988) 339–356.
[36] S. Siami-Namini, N. Tavakoli, A. S. Namin, A comparison of arima and lstm in forecasting time series, in: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), IEEE, 2018, pp. 1394–1401.
[37] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins, et al., Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai, Information Fusion 58 (2020) 82–115.
[38] I. E. Livieris, E. Pintelas, P. Pintelas, A cnn–lstm model for gold price time-series forecasting, Neural computing and applications 32 (23) (2020) 17351–17360.
[39] I. Guyon, G. Cawley, G. Dror, A. Saffari, Hands-On Pattern Recognition Challenges in Machine Learning, Volume 1, Microtome Publishing, Brookline, Massachusetts, USA, 2011.
[40] L. Erhan, M. Ndubuaku, M. Di Mauro, W. Song, M. Chen, G. Fortino, O. Bagdasar, A. Liotta, Smart anomaly detection in sensor systems: A multi-perspective review, Information Fusion.
[41] S. Lloyd, Least squares quantization in pcm, IEEE transactions on information theory 28 (2) (1982) 129–137.
[42] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T.-Y. Liu, Lightgbm: A highly efficient gradient boosting decision tree, Advances in neural information processing systems 30 (2017) 3146–3154.
[43] A. Pavao, I. Guyon, N. Stéphane, F. Lebeau, M. Ghienne, L. Platon, T. Barbagelata, P. Escamilla, S. Mzali, M. Liao, S. Lassonde, A. Braun, S. B. Amor, L. Cucu-Grosjean, M. Wehaiba, A. Bar-Hen, A. Gogonel, A. B. Cheikh, M. Duda, J. Laugel, M. Marauri, M. Souissi, T. Lecerf, M. Elion, S. Tabti, J. Budynek, P. Le Bouteiller, A. Penon, R.-D. Lasseri, J. Ripoche, T. Epalle, Aircraft Numerical "Twin": A Time Series Regression Competition, in: ICMLA 2021 - 20th IEEE International Conference on Machine Learning and Applications., Pasadena / Virtual, United States, 2021. doi:10.1109/ICMLA52953.2021.00075.
[44] S. M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, Advances in neural information processing systems 30.
[45] D. P. Kingma, M. Welling, Auto-encoding variational bayes, arXiv preprint arXiv:1312.6114.