International Forum on Aeroelasticity and Structural Dynamics
IFASD 2024
17-21 June 2024, The Hague, The Netherlands

FEEDFORWARD BACKPROPAGATION ARTIFICIAL NEURAL NETWORKS APPLIED TO DYNAMIC LOADS OF A MILITARY TRANSPORT AIRCRAFT

GE. Buendía, BS. Pulido, MA. Torralba, M. Reyes, F. Arévalo and H. Climent

Structural Dynamics and Aeroelasticity Department

Airbus Defence and Space

Avda. John Lennon s/n, 28906 Getafe

Spain

Keywords: Neural Networks, Dynamic Loads, Data Analytics

Abstract: Artificial neural networks (ANN) are known for solving complex problems and detecting nonlinear relationships between the variables of a database in a fast and accurate manner. Moreover, their storage memory optimization makes them an attractive tool for aeronautical applications such as flight control algorithms, health monitoring of structural components and flight simulators [1]. In order to study the feasibility of applying feedforward backpropagation networks to dynamic loads problems, two applications on a military transport aircraft have been analysed:

1. Assessment of potential aircraft overloading in hard landings events, where classification neural networks were used.

2. Prediction of Fatigue continuous turbulence loads, where regression neural networks were used.

The following procedures have been explored for the proper development of these neural networks:

1. Exploration of the database inputs and outputs. This includes an initial assessment of the relevance of the input parameters, the selection of the suitable outputs to be monitored and the identification of the densest regions in the database.

2. Neural training and hyperparameter tuning using Keras/TensorFlow 2.0 [2]. Sensitivity studies are performed to select the combination of the parameters that specify the details of the learning process which provide the best results, either for predicting a single or multiple outputs simultaneously. The definition of the cost function and metrics is of special relevance.

3. Interpolation and extrapolation capabilities assessment. Evaluations on the ability of the trained networks to predict results from inputs not used in the training, which are inside or outside the limits of the variable space in which they have been trained, were performed.

4. Performance evaluation with recorded flight data. Influence of the errors coming from conservative estimations of the neural networks on the fleet operations is evaluated.

The extension of this methodology to other dynamic loads problems such as fatigue discrete gust, taxi loads, and other overloading events will be explored in the future.

1 INTRODUCTION

A military transport aircraft experiences dynamic loads through its full life cycle. In- or post-flight monitoring of these loads are usually needed to support in-service events (severe turbulence, hard landing, etc.) or fatigue life consumption among other applications. A reliable and accurate loads monitoring is sometimes computationally expensive and ANNs based on calculated loads databases are expected to reduce the computational cost while keeping high reliability.

Main objectives of this paper are:

1. To explore the quality of the loads databases of dynamic landings and continuous turbulence for their usage with ANNs techniques.

2. To develop ANNs-based fast assessment tools for dynamic landing classification and continuous turbulence loads prediction using neural networks.

3. To design the generation of significant testing cases.

Section 2 explains the methodology common to both dynamic landing and continuous turbulence applications, covering the database quality exploration and the ANNs training process. Section 3 and 4 address the hard landing detection application and the fatigue turbulence loads prediction, respectively, describing the particularities of the methodology applied to these cases. Finally, Section 4 summarizes the conclusions extracted from both applications and defines future tasks where ANNs could be used.

2 METHODOLOGY

The process for both dynamic landing and continuous turbulence applications is common and composed of the following three steps:

1. Exploration of the quality of an already existing database.

2. Training the ANNs with this database to generate a simplified model.

3. Validation of the ANN-based model with artificial or real testing cases.

2.1 Database exploration

Two of the key drivers of the final performance of the neural networks are the quantity and the quality of the data used for the training process. Since the method is part of a supervised learning

process, both input and expected outputs need to be provided during learning. A bad distribution (sparsity) of the samples or the lack of data could lead to unproper training. To avoid future problems, it is key to identify this data scarcity and propose algorithms to populate the database or even regenerate it based on the findings. In this paper some techniques applied for database exploration are described:

- Histograms give a graphical representation of the independent distribution of each input parameter and the extreme values.

- Pearson correlation coefficient [3] measures linear dependence between two random variables. This metric would give a value between -1 and 1 that measures the strength and direction of the relationship. Value -1 means perfect anticorrelation, 0 no correlation at all and 1 for perfect correlation. This coefficient only works well under linear relationships between variables, and it is given by the expression

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2 \sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{1}$$

  where x_i are the values of one of the input parameters in a sample, $\bar{x}$ the mean of the values of the input parameter, y_i the values of the output in a sample, $\bar{y}$ the mean of the values of the output and n the number of points in the database.

2.2 ANN training and hyperparameter tuning

In general, dynamic loads problems are known to have an inherent complex physics, present non-linearities and require numerical simulations to solve them. The main goal of the ANNs is to create loads-prediction simple models with minimum penalties on the accuracy of the results. The branch of machine learning used for this goal is *supervised learning*. The model is trained by adjusting ANNs internal weights to obtain the desired outputs knowing the inputs.

The type of artificial neural network chosen is feedforward backpropagation [4] which can handle complex datasets and capture nonlinear relations. The algorithm (Figure 1) consists of defining an objective function that would measure the distance between the predicted output and the desired output, also known as loss function. This function is used by the optimizer to adjust the weights of each layer of the neural network in a direction that minimizes the loss. In the training loop the weights are randomly initialized and several iterations are needed to reach the weights that minimize the error. However, having the minimum error between predicted and desired outputs does not guarantee the neural network is able to generalize and perform well on unseen data.

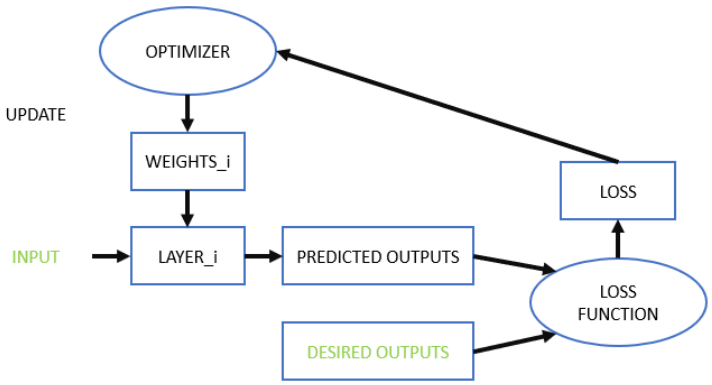

Figure 1: Feedforward backpropagation algorithm

There are parameters that can be tuned to obtain the optimal ANN performance, known as hyperparameters, and can be grouped in two categories:

ANN design parameters: activation functions, number of neurons and number of layers.

ANN training parameters:

- Distribution of the *training*, *validation* and *test* sets. The *training set* includes all the database points used to update the ANN weights. The *validation set* includes all the database points used to assess, and consequently stop, the training process. Finally, the *test set* data includes any database point that does not participate neither directly nor indirectly in the training process. This set acts as the final and most reliable evaluation of the performance of the ANN.

- Loss function, i.e., the function to be optimized. Some of the most common loss functions for regression problems are MAE (Mean Absolute Error) and MSE (Mean Square Error) and, for classification problems, the Cross Entropy Loss function is usually used.

- Optimizers. They set the strategy to change the weights and the learning rate to minimize the loss function. For this work, Adam [5] (Adaptive Moment Estimation) is selected; this algorithm is a mixture of two other algorithms:

    - AdaGrad (adaptative gradient) [6], which improves the optimization of the loss function, adapting to the difference in curvature that may exist in the different directions of the objective function. It eliminates the need to adjust the learning rate, since it updates itself during the process and is specific to each of the features, thus eliminating one of the hyperparameters to be set.

    - RMSProp (Root Mean Square Propagation) [7]. Its mathematical formulation makes the gradients of past iterations less relevant than the gradient of the current iteration. Thus, the oldest gradients are eventually forgotten. This way, the term that updates the weights does not get smaller and smaller in each iteration, which avoids the algorithm to stop updating.

- Batch size. Sample size to get the gradients between iterations during training. If computational resources are limited, the ANN can be trained using mini-batches and

predict the update on the weights and biases before composing a final value for a global update.

- Number of epochs, i.e., the iterations of the training process. The choice of this hyperparameter is one of the most complex selections, since it is difficult to know beforehand the number of epochs that will be necessary to train the network correctly. For this reason, *callback* functions are used, which are called repeatedly during the training of the ANN and are responsible for ending the ANN training process at the most appropriate point. The *callback* function chosen is the *Early Stopping* algorithm, which stops the iterations before performance starts decaying.

In addition to ANN training parameters tuning, some good practices [8] are followed during the training process:

- Reorganize the training set so successive samples are rarely in the same region or belong to the same class.
- Normalize the input data to avoid privileged directions.
- Decorrelate input variables to avoid redundant training on dependent variables.
- Weights and biases initialization to random variables to avoid initial activation function saturation and increase the learning speed.
- Stop the training process whenever a minimum error in the validation set is reached (see Figure 2). Overfitting means the network memorized the training data and underfitting that the capacity of the network is not enough to minimize even the training error.

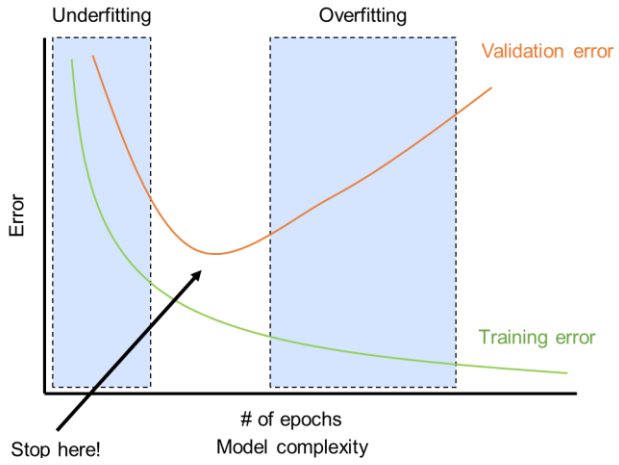

Figure 2: Training and validation errors evolution

3 APPLICATION 1: HARD LANDINGS LOADS ASSESMENT

Dynamic landing is a manoeuvre where the vertical velocity of the airplane is quickly reduced to zero when the wheels strike the ground. The process can be summarized as a transfer of kinetic and potential energy of the sinking airplane to internal energy in the shock absorption system,

where it is finally dissipated. This rapid change in velocity, and thus in application force, excites the lower vibration modes of the structure. Therefore, the structural dynamics characteristics of the structure must be taken into account.

The severity of a monitored load is determined using the criticality factor CF, which is defined as:

$$CF = \frac{Monitored\ Load}{Limit\ Load} \tag{2}$$

A monitored load is said to be critical whenever $CF \geq 1$ and supercritical when $CF \geq 1.25$, which corresponds to component plastic deformation. A hard landing will occur if any of the monitored loads is at least critical. The objective of the ANNs is to detect whether a hard landing has occurred and identify the affected components.

Note that, for this application, knowing the exact value of the *CF* is not critical but detecting precisely if it exceeded the limit threshold. Too conservative predictions imply aircrafts that did not suffer any damage to be grounded, and too lax predictions imply damaged aircraft flying. There must be a compromise between them, driven by the industry requirements. In this matter, the second case is the most important one. No operator is willing to fly with an unknown damaged aircraft, but they are willing to sacrifice a very small percentage of flights close to be critical to be declared as critical in a false alarm.

3.1 Dynamic Landing database exploration

The database explored for dynamic landings has around 90000 cases with seven inputs: (1) landing weight, (2) fuel weight, (3) load factor, (4) sink rate, (5) pitch angle, (6) roll angle and a (7) signal for spoilers. The output of this database is the criticality factors of 35 selected monitored loads and the incremental load factor of the landing. Inputs and outputs are depicted in Figure 3.

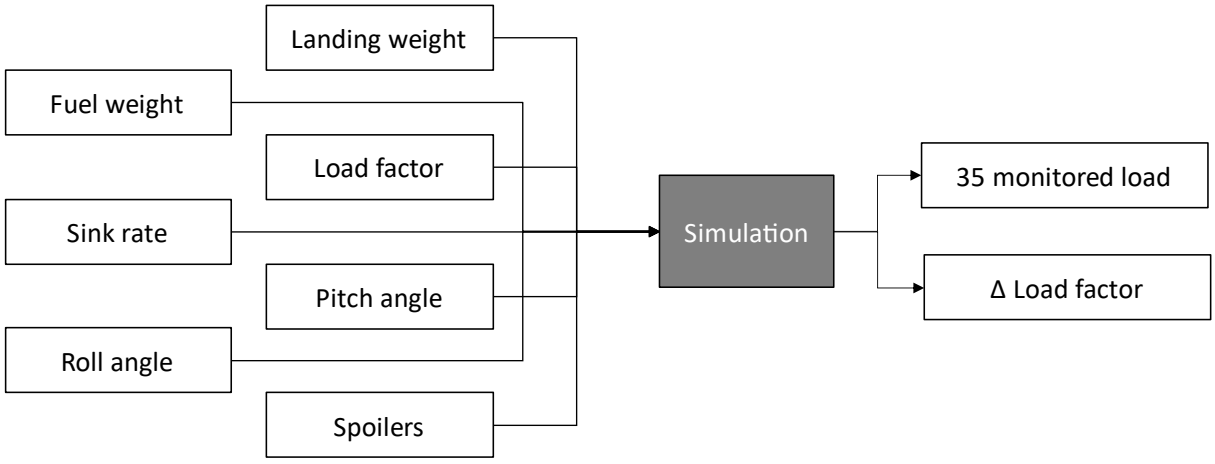

Figure 3:Inputs/Outputs from the database

Figure 4 depicts the distribution per component of these critical stations. Wing and engines are usually the most affected components in dynamic landing events.

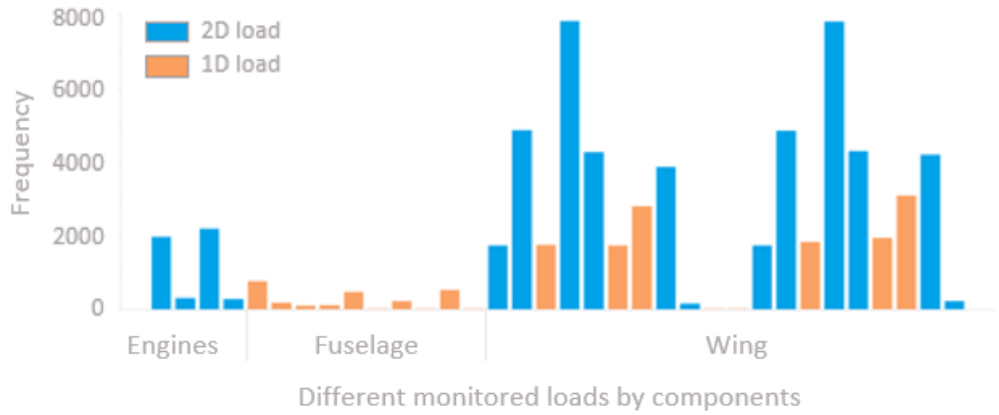

Figure 4: Frequency of critical station loads by aircraft component

The histogram distribution of the input parameters of the database is depicted in Figure 5. With these individual histograms it is observed that the main input parameters are fairly covered. However, this does not imply that all the possible combinations between parameters are addressed. Note that when there is a low load factor, the frequency of appearance is almost double to high load factors. This is because this database was designed to account for dynamic landing rebounds. In rebounds, most of the lifting force is destroyed by the activation of the spoilers during the first impact, making low load factors to appear. In this case, the database accounts for the cases with and without spoilers for low load factors.

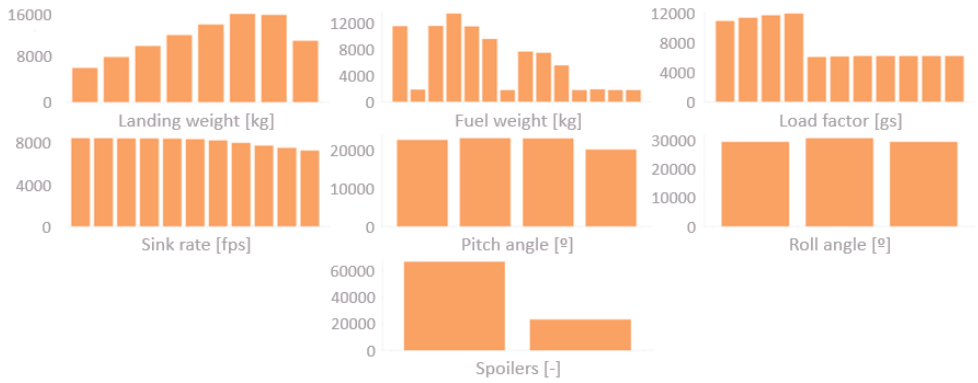

Figure 5: Dynamic landing database input parameters histograms

In order to define a measure of uncertainty of the database, at each sample the sensitivity to changing all input parameters one unit was computed as the sum of squares of the individual components. Since the unit parametric change could be changed forward or backward, the one producing the maximum ΔCF is chosen. The global change in CF for each sample is then defined as:

$$\Delta CF_{GLOBAL} = \sqrt{\Delta CF_{LW}^2 + \Delta CF_{FW}^2 + \Delta CF_{LF}^2 + \Delta CF_{SR}^2 + \Delta CF_{PA}^2 + \Delta CF_{RA}^2 + \Delta CF_{SP}^2} \quad (3)$$

The error is assumed to be:

$$\Delta CF_{ERROR} = \frac{\Delta CF_{GLOBAL}}{2} \tag{4}$$

which accounts for the worst uncertainty that could be achieved when trying to predict the output of an interpolated point. If this error is large, it means the ANN-based interpolation capability has great uncertainty. For a linear interpolator near the region of interest (*CF* ~ 1), and analyzing the most critical *CF* of every point, an average error of 20% is expected with the current Dynamic Landing database density. This shows that a regression neural network may present similar problems to predict exact *CFs*.

Due to high uncertainty when measuring the sink rate, this parameter was substituted by the increase in the load factor at the moment of impact. For a certain wing monitored load, the following Pearson correlation coefficients were obtained and shown in Figure 6.

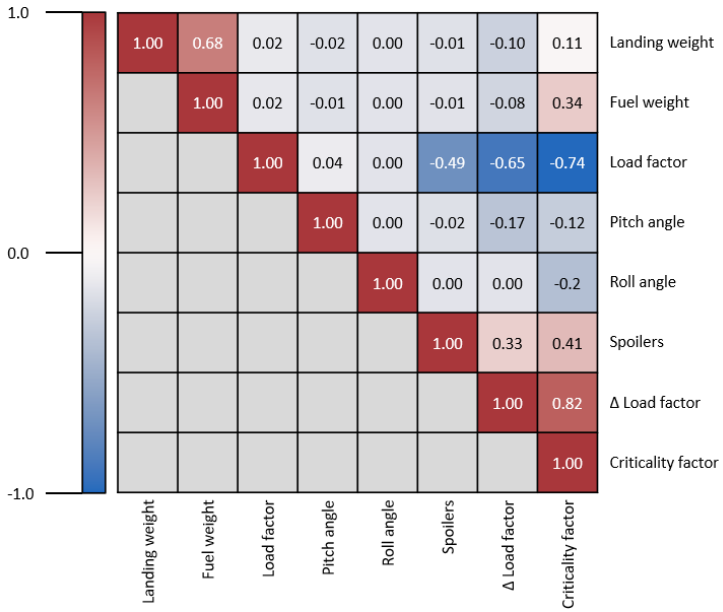

Figure 6: Inputs/output Pearson correlation coefficients heatmap for a wing monitored load

As a general trend, input parameters present little correlation between them. Having inputs as uncorrelated as possible is desirable to avoid redundancies that may decrease the performance of the ANNs. Note that the landing weight and fuel weight are correlated parameters, but it is key to have them both since the same landing weight could be achieved with different fuel weights, which are critical for the wing loads. Spoilers are inversely correlated to load factor, which account for the rebound cases where the load factor is low. Incremental load factor is strongly correlated to the load factor. However, both of them are needed to account not only for the potential energy but the kinetic energy of the impact.

Finally, note that all the inputs contribute to some extent in the criticality factor of this wing monitored load. Landing weight and pitch angle are related to the magnitude of the pintle loads that excite the dynamic response, the roll angle determines the lateral component of this excitation,

both fuel weight and the spoilers determine the dynamic response of the wing, and the load factor related inputs drive the total energy of the impact.

3.2 ANN training and hyperparameter tuning (Regression vs Classification)

The monitored-loads criticality factors CFs can be assessed by two approaches:

1. Quantitative description of the criticality, by predicting the numerical values of the CFs using *regression-type ANNs* or

2. Qualitative description of the criticality, by classifying the load criticality using *classification-type ANNs*, which groups the CFs in three different classes according to their numerical values.

For the first approach, the results of the regression ANN are depicted in a 45° scatter plot, shown in Figure 7, that compares the target outputs and the predicted outputs of the ANN in terms of *CF*. False positives are defined as the points that are not critical in reality, but are predicted as critical by the ANN, whereas the opposite applies to false negatives. The variability of the predictions is large and, to reduce the number of false negatives to zero, the *CF* output of the ANN should be shifted 0.4 units as depicted in Figure 7. This would force to declare stations with *CF* = 0.6 as critical, increasing dramatically the number of false positives.

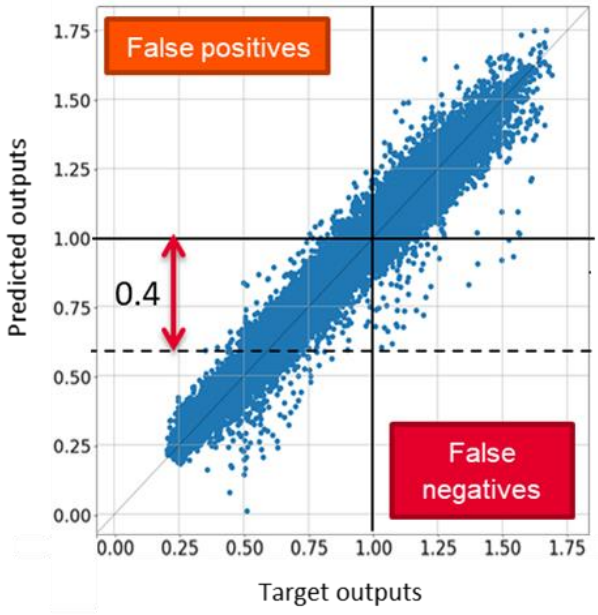

Figure 7: Regression neural network performance in training

For the second approach, the results of the classification ANN are depicted in the form of histograms depending on the pairs predicted-target output. The classification is divided in three classes: class 0 when $CF < 1$, class 1 when $1 \leq CF < 1.25$ and class 2 when $CF \geq 1.25$. This leads to 9 possible different combinations of predicted-target outputs, which are labeled as (predicted class)/(target class). Figure 8 depicts histograms on *CF* of all the possible combinations, being the

first row the correctly classified cases (00, 11 and 22), the second row the false negatives (01, 02 and 12) and the third row the false positives (10, 20 and 21).

The presented case has a displaced classification threshold of 10% in *CF*, which means that most of the cases above *CF* = 0.9 will be declared as critical. However, note that with just this 10% adjustment, there is only one false negative 01 during training.

Figure 8: Classification neural network performance in training

The **classification ANNs** has been **chosen over regression** ANNs for several reasons:

- The variability in the predictions. For regression networks even if training, validation and testing mean average prediction errors could be below 10% criticality ($\pm$0.1 in *CF*), there were outliers ranging up to 40% variability in criticality factor ($\pm$0.4 in *CF*). This would mean that a station with *CF* = 0.6 would need to be declared as critical (*CF* = 0.6$\pm$0.4), which is unacceptable. For classification network, the risk lies on the misclassification, which could be improved, although accuracies in classification over 90% were obtained with simple networks.

- Improvements to enhance the accuracy on classification ANNs can be easily implemented. To improve the previous prediction on the regression, it would be needed either to displace the outputs 0.4 in the conservative sense or retrain with more density of the database. To improve the classification ANNs, two strategies can be considered:

    - The frontier between classes could be displaced, thus increasing the amount of training data for each class. The weights assigned to each of the classes could be changed or a secondary independent artificial station could be used to make the classification more robust.

    - Although the three classes indicate if the event is non critical, critical or supercritical, it could be possible to reduce the classes into non critical and critical

(binary classification) in case not enough density of data is found for training three classes.

The architecture decided for the classification ANN was a shallow neural network with two hidden layers and ReLU activation functions with 32 neurons each.

The sets distribution was chosen to be ~90% training, 20% of the region of interest for validation, and 5% for testing as depicted in Figure 9, where the three classes 0 (CF<1), 1 ($1 \leq CF < 1.25$) and 2 ($CF \geq 1.25$) are also shown. The region of interest is defined as a region with criticality factors (CF) between 0.9 and 1.25, the purple rectangle intersecting classes 0 and 1 in Figure 9. This validation region is used to find the optimal point during the training process. The optimal point for this application is composed by the weights and biases of the ANN that minimize the classification error in this region.

Figure 9: Training, validation and test sets distribution

Adam optimizer and full sample size (batch) were considered for the training process.

In the hard landing database, there is a high predominance of class 0. However, this tool is being developed for hard landings, and class 1 and 2 are expected to be properly captured. A personalized cost function is created based on the categorical cross entropy (CCE) [9]. An inverse weighting term was implemented to give the same importance to all of the classes. This guarantees that the borders between classes are well defined and thus, misclassifications tend to be minimized. The weighted categorical cross-entropy (WCCE) was formulated as follows

$$WCCE = \sum_{i=1}^{N} -\frac{\frac{1}{n_i}}{\sum_{i=1}^{N}\frac{1}{n_i}}(y_i * \log(p_i)) \tag{5}$$

where N is the number of classes, y_i is the real probability of belonging to class i (assuming either 0 or 1 value), p_i is the probability (expressed in [0,1] interval) given by the neural network to belong to class i and n_i as the number of samples in class i. Note that the function is minimized whenever the probability predicted by the network to belong to the correct class is high ($\log(1) = 0$).

For those monitored loads where the supercritical region (class 2) has little data to train, it was chosen a binary classification network instead. Binary classification has a cost function named binary cross-entropy (BCE) [10] that takes already into account the different data density.

$$BCE = \frac{1}{N}\sum_{i=1}^{N} -(y_i * \log(p_i) + (1 - y_i) * \log(1 - p_i)) \quad (6)$$

The main goal of the ANN can be summarized into having as much correctly classified values as possible. 100% accuracy is not feasible, since the borders are the most confusing regions of classification for these networks. At the borders, there is usually a high incidence of false positives on the left part and false negatives on the right part, that decrease when moving far from it.

Figure 10 shows the frontier of critical values needs to be moved to *CF* = 0.9 to avoid misclassification on *CF* > 1. This inherently implies a sacrifice in terms of false negative declarations of the values between 0.9-1.0, but also false alarms to the left of the border.

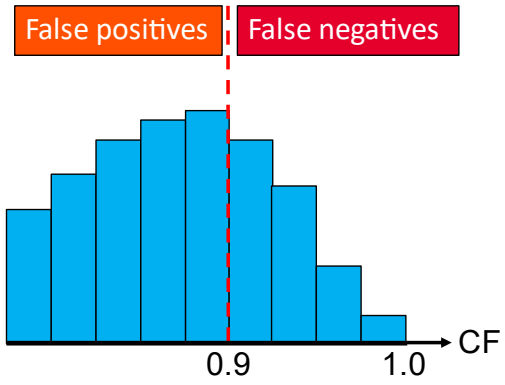

Figure 10: Misclassifications at frontiers between classes

This frontier displacement varies depending on the monitored loads. However, after the training process, most of them showed that a displaced threshold of CF = 0.9 was enough to have null false negatives over CF = 1.01. Table 1 summarizes the percentage of false positives coming from at least one of the monitored loads when setting the frontier at *CF* = 0.9.

Table 1: Average training percentage of false positives in different *CF* intervals

0.70-0.75	***0.75-0.80***	***0.80-0.85***	***0.85-0.90***
0.4%	4.0%	25.2%	69.9%

This trend is considered acceptable since the decay occurs quickly. Around 30% of the cases are properly predicted between 0.85-0.90 and around 75% between 0.80-0.85. The remaining intervals present a negligible percentage of false positives. In terms of false negatives, none appeared above 1.01 in any of the sets.

3.3 Interpolation and extrapolation capabilities

Once neural networks are compliant with the cases in the database on the three sets, a more detailed testing with new artificial data is performed near the region of interest.

The interpolation points are chosen to be intermediate points in every input parameter. The aim of these points is to check the nullity of false negatives above the limit *CF* when the network interpolates. 10000 points are generated between *CF* = [1.01,1.25].

The extrapolated points are generated to cover specific directions: lower sink rate, lower load factor, and lower and higher roll angles. The aim of these points is to check the extrapolation performance of the network including false positives and false negatives. 10000 points are generated between *CF* = [0.65,1.35].

Linear interpolation and extrapolation strategies are followed to have an estimated *CF*. If the estimated *CF* lies within a certain range defined by the user, then the point is valid and the simulation will be performed to get the exact *CF*.

The distribution of these testing sets is shown in Figure 11. Note that for the interpolation cases, around 1000 cases gave simulated values between $0.59 \leq CF \leq 1.00$, which differ from the estimated *CFs*. For the extrapolated cases, they follow the estimated *CF* distribution except for 100 cases over $CF \geq 1.35$ in the simulations.

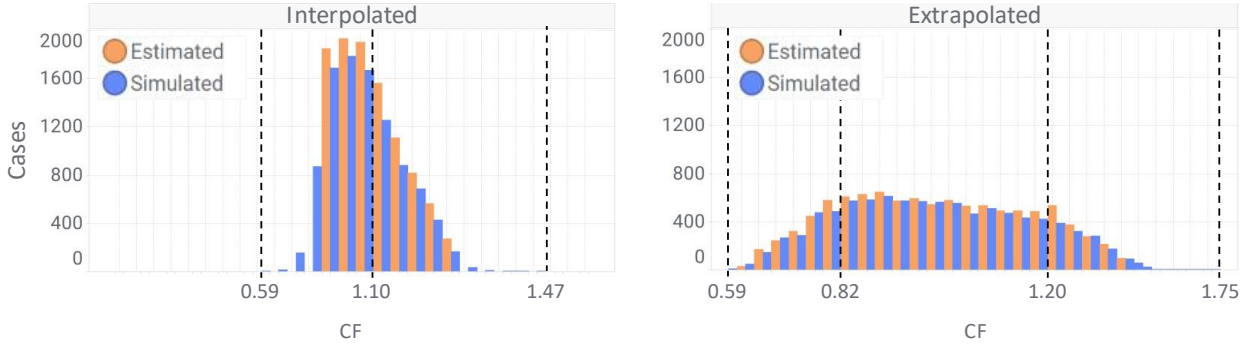

Figure 11: Interpolation (left) and extrapolation (right) points distribution with estimated and simulated maximum CF

The overall summary of misclassifications is shown in terms of confusion matrices: the target class is stated in columns and the predicted class in rows. Figure 12 (left) shows the overall confusion matrix of the interpolation points. The count refers to the overall aircraft criticality, which means the maximum criticality obtained in any of the monitored loads. Note that 82% of the interpolated points are properly classified as critical. Around 11% are falsely classified as critical, which is the result of having around 1000 points in simulation between $0.59 \leq CF \leq 1.00$ and displacing the classification threshold to 0.9. Finally, there is 15 points classified as false negatives. Most of them were landings with high kinetic energy but very high load factor, making the network struggle with the predictions of these limit cases. In fact, the probabilities of the most critical monitored loads were 60% and 40% of belonging to non-critical and critical regime, respectively. To correct for these 15 cases without affecting the false positives, the increment in load factor was increased a 5% to be conservative.

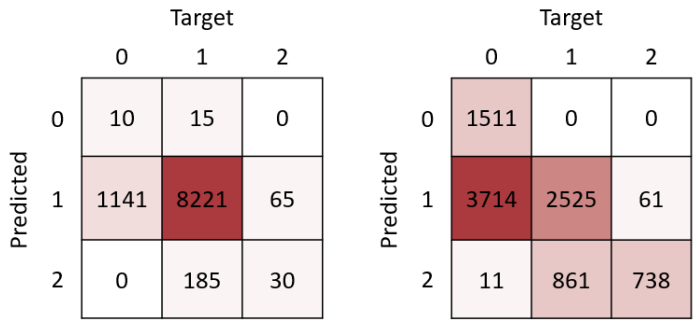

Figure 12: Interpolation (left) and extrapolation (right) points confusion matrices

Figure 12 (right) shows the overall confusion matrix of the extrapolation points. In this case extrapolation does not present any false negatives and the false positives are concentrated in the expected regions (CF = [0.85,1.01]). The 11 cases appearing in predicted 2 and target 0 combination are all above CF = 0.92 and in limit parametric conditions. They are treated as expected false positives, since their class was expected to be 1 due to the displaced threshold to 0.9. For all other points the conclusion is the network extrapolates properly within the region of interest.

3.4 Hard landings in-service events

The performance of the ANNs has been tested in two applications:

- Severity identification of the dynamic landings of the entire fleet over 2021.
- Identification of the affected aircraft component in particular hard landing events.

3.4.1 Severity identification of the fleet dynamic landings over 1 year

Classes 0, 1 and 2 defined in the ANNs are labelled as *Load Severity Indexes* (*LSIs*) in the argot of hard landing events. The order of magnitude of yearly hard landings is 10^4, whereas the order of magnitude of hard landings received that year is 10^0. Therefore, all landings predicted with an $LSI > 1$ by the ANN are false positives. $LSI = -1$ means the ANN inputs were beyond the validated limits, and thus could not be computed. They are automatically classified as severe due to the uncertainty.

Table 2 depicts the most critical LSIs predicted by the ANNs for all the landings occurring over a year. No false positives appear on landings without rebounds. However, a 0.3% of those landings are considered uncertain due to landing parameters outside the limits and would need to be simulated.

The set of landing with rebounds presents ANN-predicted false positives $LSI > 1$ due to the severity of the landing parameters. For rebounds, spoilers are active and the aircraft is not in balanced conditions with lift vs. weight ratio $L/W < 1$. This creates an extra potential energy contribution that makes the mechanical energy of the second impact more severe.

Finally, checking at the overall percentage of false positives and out of bound parameters, it is concluded that giving false alarms over 1.1% (sum of *LSI* = -1 and *LSI* > 1) of the landings over a year is a reasonable percentage.

Table 2: Overall Load Severity Index predicted by the neural networks using real landing parameters over a year

Case	Percentage	LSI = -1	LSI = 0	LSI > 1
First impacts	80%	0.3%	99.7%	0.0%
Rebounds	20%	1.2%	96.1%	2.7%
Total	100%	0.5%	98.9%	0.6%

3.4.2 Identification of the affected aircraft component in particular hard landing events

Two hard landings are shown on this paper: one without rebound, and the other one with a first impact and a subsequent rebound. The 35 monitored loads are grouped into 11 aircraft zones, and the maximum criticality factor (calculated with ad-hoc simulations) of the corresponding monitored loads is considered for each group. These calculated CFs are compared with the maximum ANN-predicted *LSI*, which are shown with 3 different colors: green, red and black, corresponding to non-critical, critical or supercritical component, respectively. Note that input parameters are normalized.

Figure 13 shows the results for the first hard landing without rebound. The overall severity of this landing is classified as 1 (red) by the ANN. However, the simulated values show that the landing is not critical because all the *CFs* < 1. In fact, the maximum is 0.97. Recap the classification threshold is displaced and false positives are expected near *CF* = 0.9 leading to expected false positives.

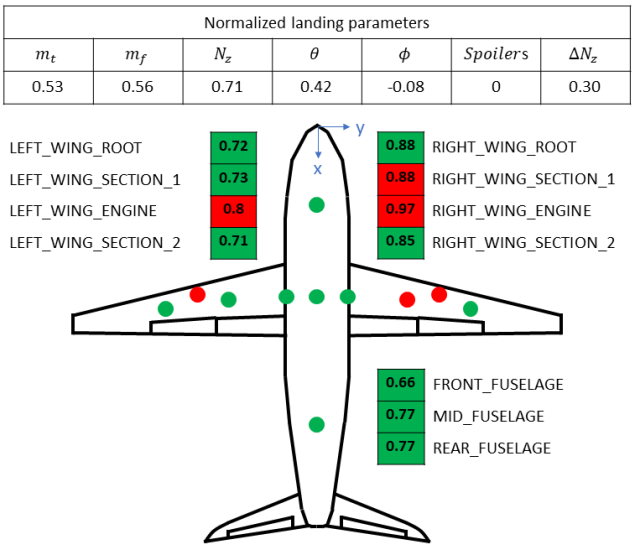

Figure 13: Landing #1 first impact simulated CFs and estimated classification LSIs

Results for the 1st impact of the second hard landing are shown in Figure 14. The overall severity of this impact is classified as 2 (supercritical) by the ANN. The maximum calculated *CFs* is 1.29 and all of the components are classified critical (red background) or supercritical (black background) by the ANN.

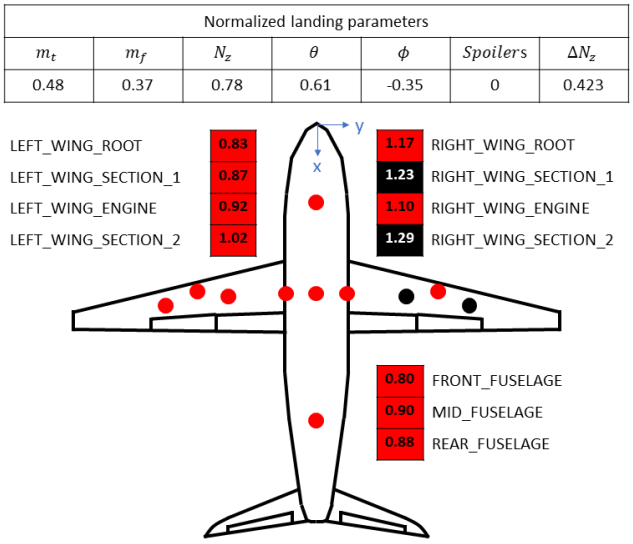

Figure 14: Landing #2 first impact simulated CFs and estimated classification LSIs

Results for the rebound of the second hard landing are shown in Figure 15. The overall severity of this landing is classified as 2 (supercritical) by the ANN. The only misprediction occurs in the right-wing section 2 where the calculated criticality factor (CF) is 1.44 and the ANN prediction is *LSI* = 1. However, it was stated that false negatives appearing between critical-supercritical classes are acceptable.

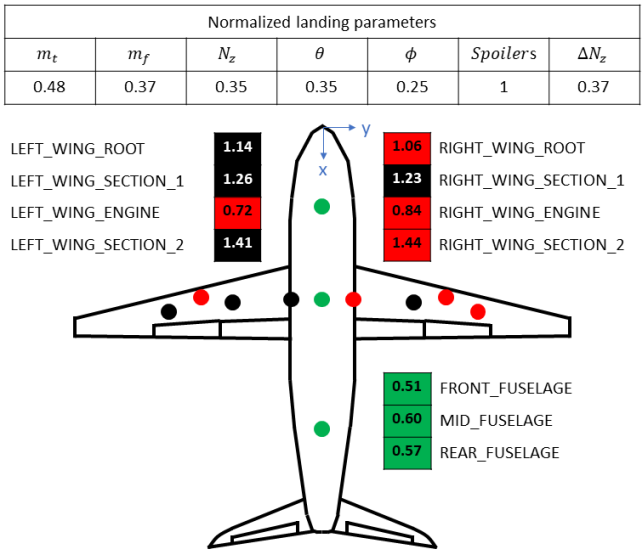

Figure 15: Landing #2 rebound simulated CFs and estimated classification LSIs

Overall, NNs estimated values of LSIs are in good agreement with the simulated results of CFs. Classification ANN show an adequate performance for the detection of hard landing events: a reduced number of false positives appear when analyzing the whole fleet over a year, and a proper overall *LSI* prediction and identification of the affected components in real hard landings is obtained by using ANNs.

4 APPLICATION #2 : FATIGUE TURBULENCE LOADS PREDICTION

A regression ANN is applied to the prediction of atmospheric turbulence loads in different monitoring stations of the military transport aircraft. The ANN database used is the one used by the aircraft structural health monitoring system or SHMS.

4.1 Database exploration

The databases are associated with vertical and lateral turbulence events. They are analytically calculated using the mass, structural and unsteady aerodynamics models used for dynamic load certification calculations and using the Von Kármán atmospheric turbulence model. They consist of 11 input parameters, two associated to the flight point and nine to the mass state, and the forces and moments in each monitoring stations. The exploration of this database has been carried out by subjecting the input variables to a Pearson correlation study, as exposed in Figure 16, and by obtaining its histograms, shown in Figure 17 and Figure 18.

Pearson indicator shows most of the input parameters are uncorrelated. Since the number of inputs is not high, it was decided to keep all of them and test the results of the ANN using this number of inputs.

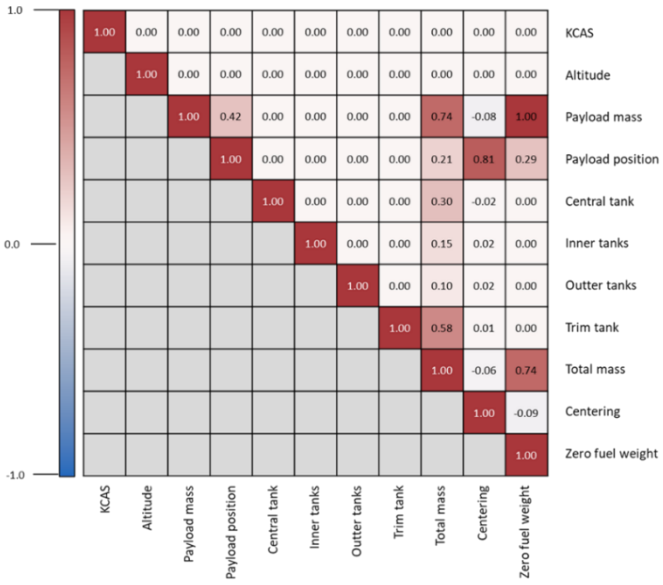

Figure 16: Pearson correlation coefficients for the input magnitudes

The histograms of the input variables show how most of the variables present a sparse behaviour. This is due to the fact that the database was designed to be used by a linear interpolator, so the intermediate values with linear behaviour can be interpolated without major problems. This will be a test of the neural network's ability to interpolate into these empty areas of the database.

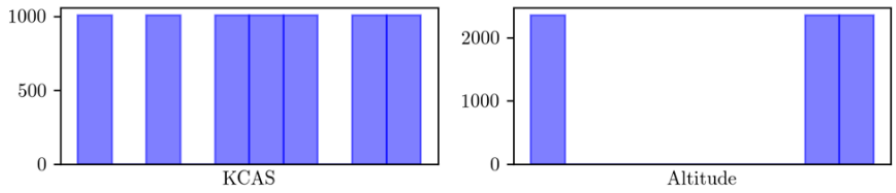

Figure 17: Flight point variables space distribution

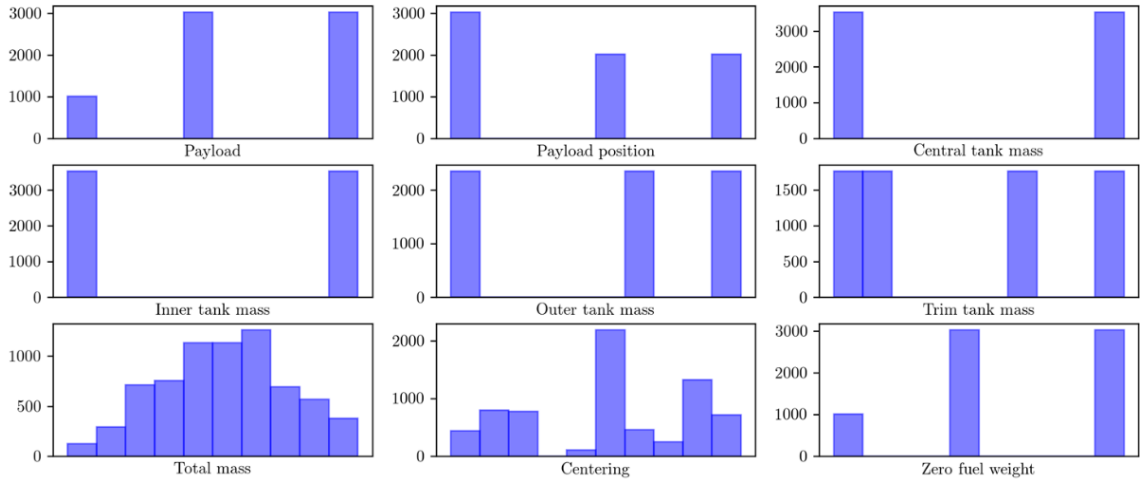

Figure 18: Mass point variables space distribution

4.2 ANN training and hyperparameter tuning

Fatigue loads are continuous values; thus, a regression neural network is employed to predict these values for a propeller-driven Military Transport Aircraft. In order to find the hyperparameters that optimize the final result, a sensitivity study of the network hyperparameters was carried out. These sensitivity studies have been performed both for an ANN that predicts a single magnitude and for an ANN that predicts multiple magnitudes.

The two main loads to which the wing is subjected are shear force and bending moment, but bending moment is more likely to be the main design criterion in the majority of wing designs, particularly important in the design of wing spars [11]. The maximum loads on the wing are seen at the wing root, so the first magnitude studied is the bending moment, in wing root for the single-output ANN and in different monitoring stations for the multiple-output ANN. It has been decided to work with 72 different hyperparameter combinations, shown in Table 3.

Table 3. Hyperparameters combination summary for the sensitivity studies

HYPERPARAMETER	SINGLE-OUTPUT			MULTIPLE-OUTPUT		
Number of hidden layers	2			2		
Neurons in hidden layers	4	8	16	8	16	32
Activation functions	Tanh		Linear	Tanh		Linear
Split ratio	70/25/5	75/20/5	80/15/5	70/25/5	75/20/5	80/15/5
Optimizers	SGD		Adam	SGD		Adam
Loss function	MAE		MSE	MAE		MSE
Metrics	MAE			MAE		
Epochs	Early Stopping			Early Stopping		
Batch size	Full sample			Full sample		

The distribution of points for training, validation and test sets is random, to ensure the points within the sets cover the entire data space and prevent training and evaluating the model in specific regions, which can lead to biased results. For that reason, the combinations have been run three times and the results of the three iterations have been averaged. The performance of the ANNs is characterize by two properties: (1) the percentage of points of the test set predicted with an error equal to or less than 5%, and (2) the average number of epochs that the network needed to perform the training. The results of these two properties are attached in Figure 19 for a single-output (wing root bending moment) ANN. It was considered adequate for the model to predict at least 90% of the test points with such error, which is indicated by the red line.

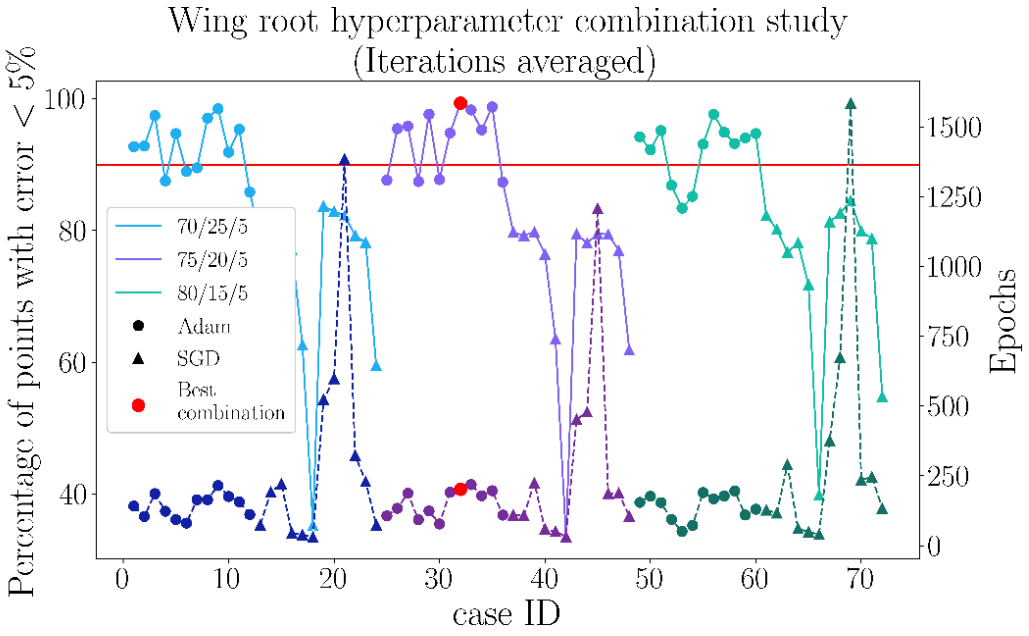

Figure 19: Percentage of points within the test set predicted with error $\leq 5\%$ (continuous) and average epochs of training (dashed) for all combinations of hyperparameters

For the multiple-output ANN, the bending moment has been studied in five different monitoring stations, but since the approach is the same as for the single-output ANN, the sensitivity study plots will be omitted. The models that have provided the best results to predict the bending moment in wing root and in different monitoring stations are presented in Table 4.

Table 4. Combination of hyperparameters of the chosen to predict single and multiple outputs

Output	Neurons hidden layers	Activation function	Optimizer	Loss function	Split ratio
Single	8	Tanh/tanh/linear	Adam	MSE	75/20/5
Multiple	32	Tanh/tanh/linear	Adam	MSE	80/15/5

The ANN configuration chosen for both single- and multiple-output problems is used to predict, not only the bending moment, but also other aircraft magnitudes, in order to test the ANN capability. In this paper we will present results, using a single- and multiple -output ANN, for:

- Wing shear force, bending and torsional moment at five different monitoring stations.
- Bending moment, lateral force and torsional moment at VTP tip and root.

To analyse the results obtained, the criteria in Table 5 are followed. First, a Kolmogorov-Smirnov test is performed to determine whether the target and predicted values come from the same statistical data distribution and are therefore similar. Secondly, the results of the ANN are plotted against the target ones around a straight line at 45°. To conclude that the network is working correctly, it is necessary to comply with both tests. An additional analysis of the distribution of the relative errors is then performed, as a method to confirm that the chosen ANN is working properly.

Table 5. Summary of the steps to be followed in the evaluation of the regression neural network model

Test	Does it meet the test?			
Kolmogorov-Smirnov	No	Yes	No	Yes
Scatter plot	No	No	Yes	Yes
Error distribution	Not applicable	Not applicable	Not applicable	Applicable
Outcome	Refine and boost data and/or model			Model works well

A brief summary of these methods is presented herein. The Kolmogorov-Smirnov (KS) test calculates the cumulative distribution function of both distributions, compares them and calculates the maximum absolute distance between them,

$$D_{mn} = max|B(x) - G(x)| \tag{7}$$

where B(X) and G(x) are the distribution of the predicted and the target values, and D_{mn} is the KS statistic. Then, this value must be compared to a threshold value, the critical KS statistic, $D_{mn,\alpha}$

$$D_{mn,\alpha} = c(\alpha)\sqrt{\frac{m+n}{mn}} \tag{8}$$

where c(α) represents the critical value of the Kolmogorov distribution evaluated at α, which is the significance level (probability of rejecting the null hypothesis when it is true). In this case, the null hypothesis states that there is no difference between two distributions, and therefore, both samples compared come from the same statistical data distribution. This null hypothesis is rejected when $D_{mn} > D_{mn,\alpha}$.

The critical value of the Kolmogorov distribution c(α) [12] is obtained by

$$c(\alpha) = \sqrt{-\frac{1}{2}ln(\frac{\alpha}{2})} \tag{9}$$

To verify the reliability of the study, it is necessary to calculate the p-value, a probability value which indicates that the results obtained are not due to chance. The aim is for this value to be as close to 1 as possible.

The second evaluation of the model will be carried out by means of a scatter plot, in which the target values are compared with the values predicted by the network. In this way, what is sought is to observe how these results are distributed around a straight line at 45°.

The evaluation of the error distribution is the third check performed, and only applies if the two previous tests have been satisfactorily accomplished. The error distribution will show how the relative error made when predicting the values of the test set with respect to the target values is distributed. The expected result of this test is that the error distribution follows a Gaussian distribution.

Once the methods to evaluate the results obtained by the ANN are outlined, results for the single- and multiple-output ANN are presented below. First, evaluation of values predicted for the bending moment at the wing root and lateral force in the VTP tip of the propeller-driven Military Transport Aircraft studied are shown in Figure 20 and Figure 21, respectively.

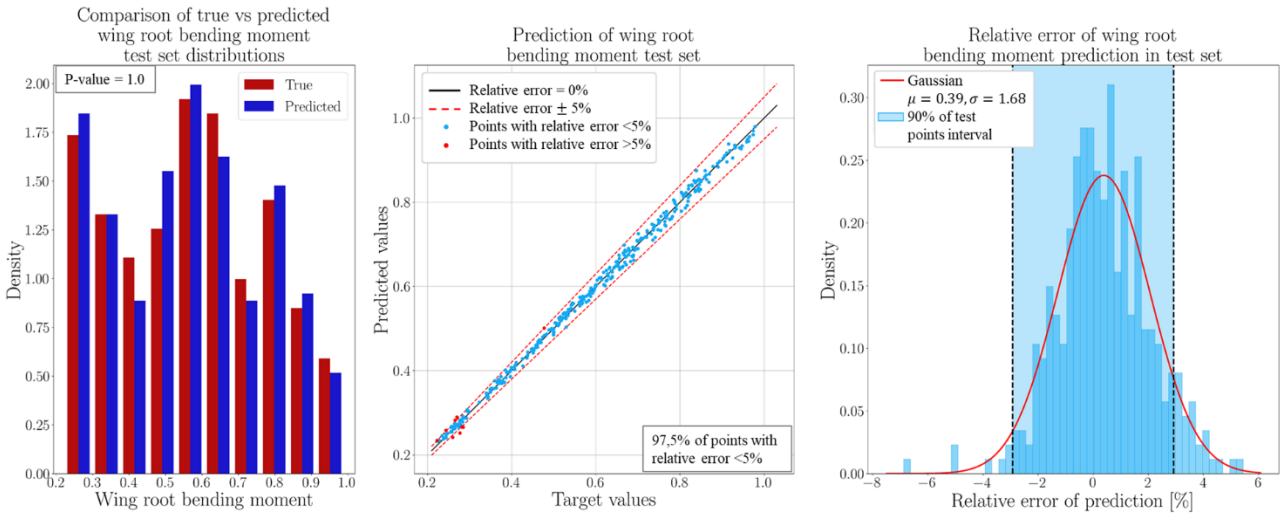

Figure 20: KS test, scatter plot and error distribution for the bending moment at wing root

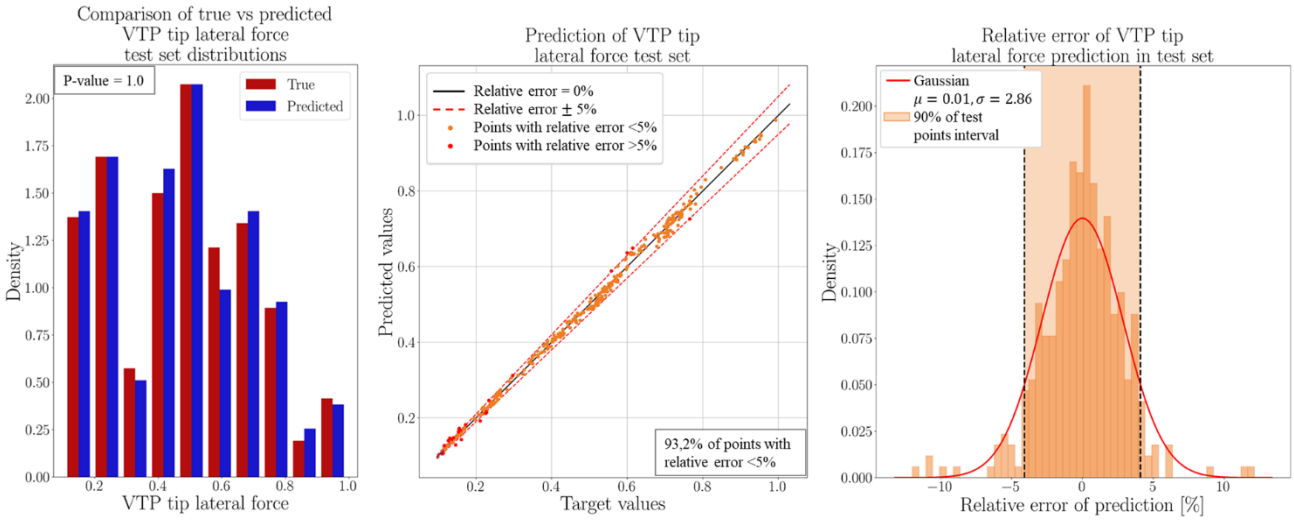

Figure 21: KS test, scatter plot and error distribution for the lateral force at the VTP tip

The value of the KS statistic for the bending moment at wing root ($D_{mn} = 0.0226$) and for the lateral force at VTP root ($D_{mn} = 0.0198$) are both lower than the critical value for $\alpha = 5\%$ ($D_{mn,\alpha} = 0.1022$). This indicates that samples of predicted and target values belong to the same statistical data distribution for both cases, and the p-value obtained indicates that this result is reliable. Thus, it is concluded that first test of the ANN evaluation is satisfied. The figure on the middle shows that 97.5% of the points belonging to the test set of the bending moment are predicted within a 5% of error margin, and it has also been calculated that 90% of the points are within a margin of error of 2.9%. For the lateral force at the tip of the VTP, this percentage takes 93.2% as a value, and 90% of the predictions lay within a 4.1% margin of relative error. The remaining magnitudes predicted by a single-output ANN for both wing and VTP are presented in Table 6 and Table 7, respectively.

Table 6. Summary of the results obtained for bending moment at each wing monitoring station (MS) using a single-output neural network

Monitoring Station	Magnitude	KS statistic	KS p-value	Test points with error < 5%	Error margin where 90% of predictions lay
Wing root	Bending moment	2.26E-02	9.99E-01	97.5%	2.9%
Wing centre		1.98E-02	9.99E-01	96.6%	3.6%
Wing tip		1.98E-02	9.99E-01	98.6%	2.7%
Wing root	Shear force	2.26E-02	9.99E-01	93.7%	4.2%
Wing tip	Torsional moment	3.33E-02	9.87E-01	98.6%	2.9%

Table 7. Summary of the results obtained for the lateral force, bending and torsional moment at both monitoring stations of the vertical stabilizer, using a single output neural network.

Monitoring Station	Magnitude	KS statistic	KS p-value	Test points with error < 5%	Error margin where 90% of predictions lay
VTP tip	Lateral force	5.66E-02	9.99E -01	93.2%	4.1%
	Bending moment	2.54E-02	9.99E-01	96.3%	3.6%
	Torsional moment	2.26E-02	9.99E-01	93.2%	3.8%
VTP root	Lateral force	1.98E-02	9.99E-01	98.6%	2.9%
	Bending moment	3.68E-02	9.70E-01	97.5%	3.3%
	Torsional moment	2.26E-02	9.99E-01	92.9%	4.1%

The prediction of the shear force, bending and torsional moment at different monitoring stations of the wing and the lateral force, bending and torsional moment at the VTP is very satisfactory, so the ANN chosen works well with magnitudes related to both vertical and lateral turbulence.

These magnitudes have also been predicted using a multi-output ANN. A total of 32 neurons in the hidden layers have been used to predict the VTP magnitudes, but a network of 48 neurons was needed to predict the shear force, bending and torsional moment in the five monitoring stations of the wing. Results of these are provided as a comparison of the margin of error within which the 90% of the predictions lay between the single-output and the multi-output neural network, presented in Table 8 and Table 9.

Table 8. Comparison of margin of error within which the 90% of the predictions made by a single and multiple output neural network lay, for the bending moment of the wing at different monitoring stations

Monitoring Station	Magnitude	Single output	Multiple output
Wing root	Bending moment	2.9%	3.5%
Wing centre		3.6%	3.6%
Wing tip		2.7%	4.8%
Wing root	Shear force	4.2%	4.5%
Wing tip	Torsional moment	2.9%	5.4%

Table 9. Comparison of margin of error within which the 90% of the predictions made by a single and multiple output neural network lay, for the lateral force, bending and torsional moment of VTP tip and root

Monitoring Station	Magnitude	Single output	Multiple output
VTP tip	Lateral force	4.1%	3.8%
	Bending moment	3.6%	2.9%
	Torsional moment	3.8%	5.4%
VTP root	Lateral force	2.9%	3.9%
	Bending moment	3.3%	3.8%
	Torsional moment	4.1%	7.2%

The multi-output ANN chosen to predict the loads at the wing and the VTP at their different monitoring stations provide good results, with the exception of the torsional moment for both components, whose prediction is slightly worse. In the case of the wing, the torsional moment had to be segregated from the shear force and the bending moment in a 16-neuron network. This may be due to the fact that the shear force (lateral force for the VTP) and the bending moment are closely related, while the torsional moment is a more independent magnitude.

It can be seen how, by predicting the magnitudes with a single-output ANN, better results are obtained in general, since the network is trained specifically for that magnitude, without having to take into account relationships between different magnitudes.

4.3 Interpolation capabilities of the ANNs. Comparison with linear interpolation techniques.

This section presents a study of the ANN's ability to predict loads at those intermediate points of the envelopes that were not defined in the databases. The aim is to observe whether the ANN has enough capacity to use a database that was originally created for using linear interpolation techniques (and not designed for ANN techniques), or whether it would be necessary to define intermediate points in these empty areas in order to generate a denser space of values in the database. For this purpose, the single-output ANN that predicts the bending moment at the wing root, stored together with the value of the weights and biases that optimize it, is used. Two cases have been studied:

- A flight speed value not used to train the network with an altitude value used for training.

- A flight speed and altitude values that have not been used to train the network.

These two new flight points were analyzed for all mass states available, which is shown in Figure 22. The blue values represent the wing root bending moment vs flight speed (left) and vs altitude (right) within the test set predicted by the ANN, while the other points are the ones predicted for the new flight conditions. For one particular mass state, the value of the predicted wing root bending moment and its comparison with the values obtained by the simulations (target values in this case) are shown in Table 10.

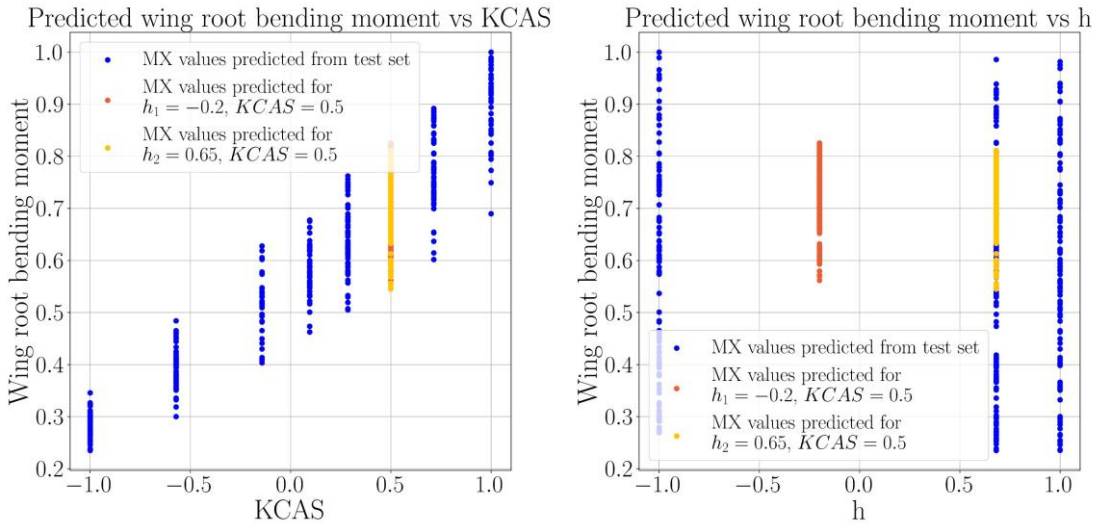

Figure 22: Prediction results for the two cases of unknown flight point for all mass states

Table 10. Prediction results for the two cases of unknown flight point for a particular mass state

Velocity	**Altitude**	**Analytical value**	**Predicted value**	**Relative error** [%]
0.5	-0.2	0.656	0.675	2.9
0.5	0.65	0.649	0.646	-0.4

The error is below 1% for the point where the flight speed value is not used in the training process but the altitude is. This error increases up to 3% for the point where both the flight speed and altitude were not used in the ANN training process. However, the maximum error made in the prediction of the loads at these points are similar than the error made by the linear interpolator (3%) for this magnitude. The advantage of the ANN over the linear interpolator is the storage memory it requires, since the number of parameters stored for the ANN operation is two orders of magnitude less than those for the linear interpolator and the complete database.

5 CONCLUSIONS

In this paper, the feasibility of using ANN to predict aircraft loads in dynamic landing and turbulence encountering events is evaluated. In general, target values (from loads analyses) are in good agreement with the predicted values obtained with ANNs.

The already existing databases were populated enough to be trained by specific ANN suited to the applications.

Classification neural networks fitted best the hard landing detection application. It was tested with fleet data over a year to show the percentage of false positives obtained is negligible with no false negatives. It was also tested with particular hard landing events to evaluate if the affected components were properly identified, being the result satisfactory.

Regression neural networks have proven to be a viable alternative to traditional tools such as linear interpolators **in predicting continuous turbulence loads**, as the errors made by both tools are comparable, with the advantage of occupying two orders of magnitude less memory in case of considering its use on board the aircraft.

The satisfactory results obtained for the described applications may imply the extension of the use of neural networks for other complex dynamic problems such as taxi loads or discrete gust events in the future.

REFERENCES

[1] W. E. Faller and S. J. Schreck, "Neural networks: Applications and opportunities in aeronautics," *Progress in Aerospace Sciences,* vol. 32, no. 5, pp. 433-456, 1996.

[2] M. Abadi et al., «TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems,» 2015. [Online]. Available: https://www.tensorflow.org/.

[3] Royal Society of London, Proceedings of the Royal Society of London, London: Harrison and Sons, 1895.

[4] F. Chollet, «Artificial intelligence, machine learning, and deep learning,» de *Deep Learning with Python,* Connecticut, Manning Publications Co., 2017, pp. 4-13.

[5] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *Published as a conference paper at ICLR 2015,* 2015.

[6] R. Gylberth, "An Introduction to AdaGrad," 03 may 2018. [Online]. Available: https://medium.com/konvergen/an-introduction-to-adagrad-f130ae871827.

[7] D. Shulman, "Optimization Methods in Deep Learning: A Comprehensive Overview," 2023.

[8] Y. LeCun, L. Bottou, G. Orr and K. Müller, "Efficient BackProp," in *Neural Networks: Tricks of the Trade* , Berlin, Springer, 2012, pp. 9-48.

[9] I. J. Good, "Rational Decisions," *Journal of the Royal Statistical Society. Series B (Methodological),* vol. 14, no. 1, pp. 107-114, 1952.

[10] D. R. Cox, "The Regression Analysis of Binary Sequences," *Journal of the Royal Statistical Society. Series B (Methodological),* vol. 20, no. 2, pp. 215-242, 1958.

[11] G. Ciurpita, "Bending moments," 06 September 2002. [Online]. Available: https://ciurpita.tripod.com/rc/rcsd/bendMom/bendMom.html.

[12] D. W. a. E. C. M. Hollander, Nonparametric Statistical Methods, 2015.

[13] J. Brownlee, "Machine learning mastery," 12 October 2021. [Online]. Available: https://machinelearningmastery.com/gradient-descent-with-momentum-from-scratch/.

[14] S. Taylor, "Corporate finance institute," [Online]. Available: https://corporatefinanceinstitute.com/resources/data-science/r-squared/.

[15] J. Berkson, "Tests of significance considered as evidence," *Journal of the American Statistical Association,* vol. 37, no. 219, p. 325–335, 1942.

COPYRIGHT STATEMENT