

Rotorcraft takeoff analysis and classification to detect outlier operations that could present a safety risk

Ricardo F. da Silva*, Gabriel N. Achour†, Alexia P. Payan‡, Charles Johnson§, and Dimitri N. Mavris¶
Aerospace Systems Design Laboratory, School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, 30332-0150, USA
Federal Aviation Administration, Atlantic City International Airport, NJ 08405, USA

Various reports from entities such as the Federal Aviation Administration (FAA) and the National Transportation Safety Board (NTSB) have shown a recent increase in the number of incidents involving helicopters. The versatility of rotorcraft operations makes the establishment of safety metrics challenging. Yet, flight data monitoring (FDM) programs enable the implementation of data-based models and analyses that can contribute to improving the safety of helicopter operations. Traditionally FDM programs have featured exceedance-based data analyses by defining safety thresholds. However, recent advances in data science, and more particularly in deep learning techniques, have paved the way for a more reliable definition of safety thresholds via the use of outlier detection algorithms. This paper focuses on the implementation of an anomaly detection model for the takeoff phase which represents a large portion of incidents in rotorcraft operations. After generating training data and augmenting the dataset, the takeoff segment is extracted from each flight data record. Then, the type of takeoff performed is identified through a classification algorithm, and finally, a recurrent neural network composed of long short term memory cells is implemented to detect anomalies or outliers in the input takeoff data.

I. Introduction

ALTHOUGH the rate of rotorcraft accidents has decreased in the past decade, reports from the U.S. Helicopter Safety Team (USHST) show that the rate plateaued then increased in recent years [1]. Moreover, the statistics show that the rate of both accidents and incidents per 100,000 flight hours is still higher compared to the rate for fixed-wing flights [2]. Thus, to improve the safety of rotorcraft operations, the National Transportation Safety Board (NTSB) suggested broadening the use of flight data recorders (FDR) in cockpits [3]. According to the NTSB datasets, a significant portion of non-fatal and fatal general aviation accidents occur during the takeoff phase of the flight [4]. As such, the Federal Aviation Administration (FAA) is collaborating with universities through the Partnership to Enhance General Aviation Safety, Accessibility and Sustainability (PEGASAS) to study how to utilize new sources of safety data, such as Flight Data Monitoring (FDM) to analyze these events, such as takeoffs. To detect anomalies during the different phases of flight, FDM programs typically rely on exceedance-based analyses. This means that safety thresholds must be established for each monitored flight variable. As such, when applied to takeoffs, the FDM program needs to detect each possible takeoff category, and safety thresholds would need to be established for each of these categories. However, due to the variety of rotorcraft missions, the establishment of general thresholds can represent a challenge in this field. This effort represents a continuation of ongoing rotorcraft safety research with PEGASAS and the FAA to identify patterns and establish safety thresholds from a dataset of recorded flights [5]. Since helicopters can perform a variety of takeoffs having different profiles, this effort instead focuses on takeoff analysis; particularly outlier detection from the flight data during the takeoff segment to identify and anticipate any potential incident or accident during the takeoff phase.

*PhD Student, Aerospace Systems Design Laboratory, School of Aerospace Engineering, 270 Ferst Drive, Atlanta, GA, 30332-0150, AIAA Member

†PhD Student, Aerospace Systems Design Laboratory, School of Aerospace Engineering, 270 Ferst Drive, Atlanta, GA, 30332-0150, AIAA Member

‡Research Engineer II, Aerospace Systems Design Laboratory, School of Aerospace Engineering, 270 Ferst Drive, Atlanta, GA, 30332-0150, AIAA Member

§General Engineer, System Safety Section, Aviation Research Division, NextGen WJHTC Office, Federal Aviation Administration, William J. Hughes Technical Center, Atlantic City International Airport, NJ 08405

¶S.P. Langley Distinguished Regents Professor, Boeing Regents Professor of Advanced Aerospace Systems Analysis, School of Aerospace Engineering, Director Aerospace Systems Design Laboratory, 270 Ferst Drive, Atlanta, GA, 30332-0150, AIAA Associate Fellow

II. Background

The data recorded in FDM programs corresponds to multi-dimensional time series. Outlier detection in this type of data has been extensively explored in the literature. In most cases, auto-encoders are used to identify anomalies in the data. Indeed, the bottleneck architecture of auto-encoders (AE) compacts the data into a latent vector capturing patterns within the data. The reconstruction error can help identify outliers that were filtered out in the encoding process. Convolutional neural networks (CNN) in an auto-encoding architecture have been used for multi-dimensional time series outlier detection problems [6, 7]. While CNN models are typically used for image classification, they can also deal with multi-dimensional time series data by setting up the different variables as a matrix where rows correspond to the parameters and the columns to the time increments. Zhong et al compared the performance of a 2D CNN AE model with other techniques and showed that the use of recurrent neural networks (RNN) and more precisely the usage of long short-term memory (LSTM) cells can outperform CNN based approaches [8]. Other efforts applied LSTM models to outlier detection problems. Zhong et al. proposed a spatio-temporal correlation using LSTM technique for anomaly detection and recovery prediction of UAV flight data [9]. The first step is to mine the spatio-temporal correlation in-flight data and to obtain the correlation parameter sets from an artificial neural network. After that, the LSTM model is created, and the mapping among different parameters is realized. Lastly, the mapped model is used to detect anomalies and recovery prediction. Qin et al. used Quick access record (QAR) flight data containing information of aircraft landing at Kunming Changshui International and Chengdu Shuangliu International airports as experimental data to implement a hybrid model combining a time-feature attention-based convolutional autoencoder with the HDBSCAN clustering algorithm [10]. The autoencoder is constructed and trained to extract flight features whereas the HDBSCAN works as an anomaly detector. Rey et al. created a data-driven approach to classify flights as safe or at risk [11]. A boosted tree classifier was used to conduct analysis and prediction, and in that case, leaf nodes of the tree were related to the features at intermediate nodes and a flight could be predicted as at risk or not at risk. The approach proposed could be used as a continuity of the outlier detection analysis. Ning et al. used Long Short-Term Memory network based Auto-Encoders (LSTM-AE) for complex aircraft system fault detection and classification using raw time-series data from heterogeneous sensors [12]. A model is created using LSTM-AE, then a system health index (HI), representing the nominal behavior, and fault feature vectors are derived from the reconstruction error matrix for fault detection and classification.

III. Problem definition

Before training an LSTM model and detecting outliers in the takeoff recordings, some preliminary steps are necessary to prepare the data. The first one is to extract the takeoff segment from the recorded data. Multiple definitions can exist to define the beginning and end of this flight phase. Once the takeoff segment is extracted, the next milestone is to identify the type of takeoff. The FAA helicopter handbook provides definitions and instructions for the most common types of takeoff [13]:

- Normal takeoff (from surface or from hover): taking off and transitioning into a normal climb using the minimal amount of power
- Maximum performance takeoff (from surface or from hover): climb at a steep angle using maximum available aircraft performance to clear an obstacle as soon as possible
- Rolling takeoff: moving forward on the surface gaining speed before lifting off, the only type of takeoff where the vehicle has some velocity before liftoff

In addition to the basic takeoff maneuvers, this works also includes category A and category B takeoffs. These are precision takeoffs meant to allow for safe operation in the case of an engine failure or other emergency and are also employed when obstacles or other conditions preclude one of the other takeoff profiles.

After the extraction and classification of the takeoff segment, the outlier detection process can start. A Recurrent Neural Network - LSTM was implemented and trained to create models of each takeoff category. The original datasets used were generated by a flight simulator operated by different pilots. The proposed approach is shown in Figure 1.

IV. Methodology

A. Takeoff segment identification

For this task, the beginning and end points of the takeoff segment needs to be defined. For the beginning of the segment, the liftoff point can be used. From the flight recorders, the "weight on wheels" signal can be used to determine

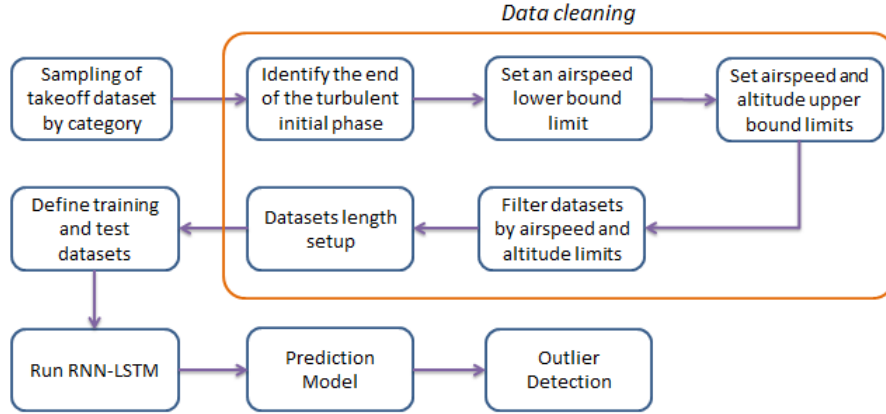


Fig. 1 Framework proposed for outliers detection in helicopter takeoffs

when the vehicle departs. If such parameter does not exist in the data, the altitude can be used to determine the instant when the vehicle takes off (i.e. wheels leave the ground or altitude or speed changes signaling departure from a hover). For the segment end, there can be different definitions such as those that are either altitude or airspeed based. For example, in one adaptation, the takeoff ends whenever the vehicle reaches 150ft altitude from the liftoff point [14]. Yet, this definition lacks accuracy as it depends on the takeoff type and the pilot, the takeoff profile can vary, hence using only the altitude can often be insufficient.

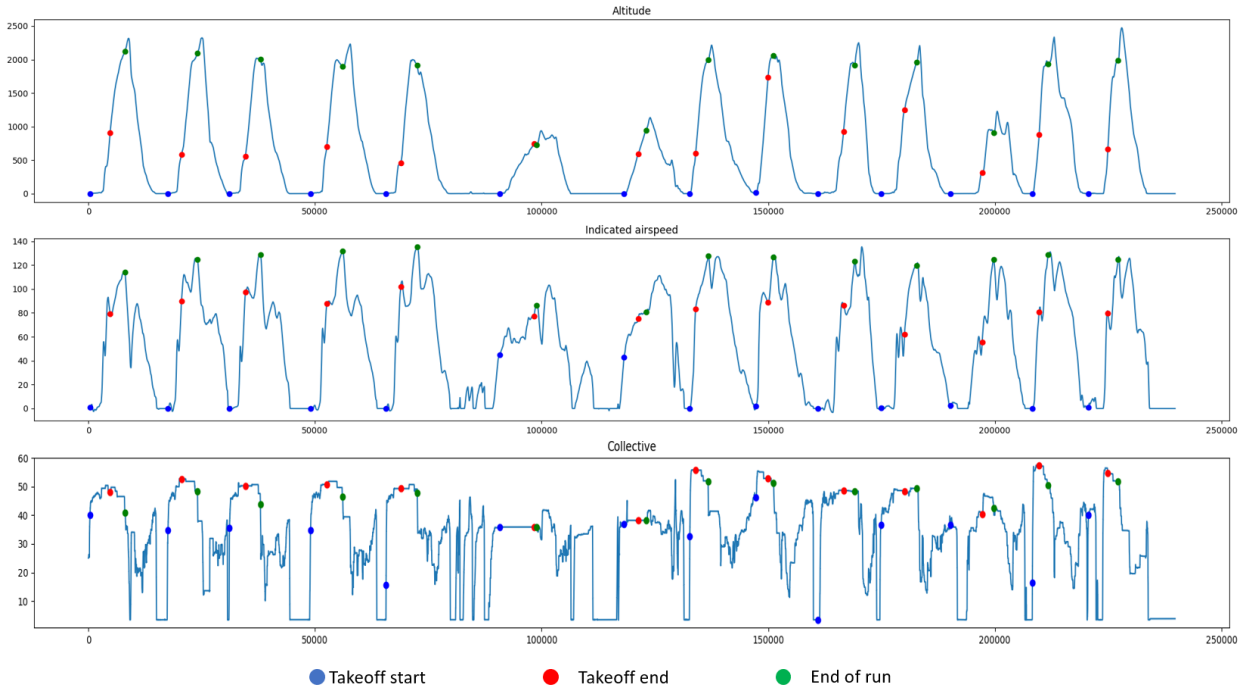


Fig. 2 Succession of takeoffs and landings with timestamps of the beginning and end of each takeoff

To identify a more suited ending point, simulator runs with different pilots on a variety of takeoff conditions were coordinated. During each flight, the pilots were asked to indicate at what point they consider the takeoff segment is ending. Figure 2 shows different metrics from the flight recorder during the succession of takeoffs conducted and their corresponding ending point. From the different graphs, it can be observed that an altitude-based definition is not accurate, but instead, the position of the collective control can provide insight. Indeed, in most cases the takeoff ends

coincides with the first peak of the collective position from the flight recorder. For rolling takeoffs, corresponding to the sixth and seventh takeoffs on the graph, the collective position does not indicate the end of the segment, instead a velocity threshold can be used. As such, these metrics, and the "weight on wheels" can determine the takeoff segment from the flight data as depicted in Figure 3.

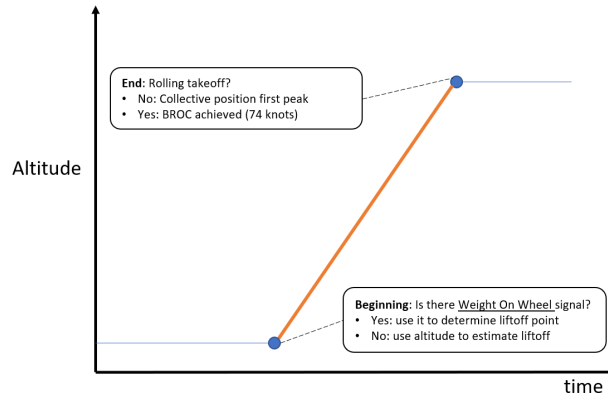


Fig. 3 Takeoff segment definition

B. Takeoff classification

The classification task was performed on the flight records generated on the simulator. Figure 4 shows the plot of the altitude versus the indicated airspeed of a subset of different takeoff types flown by different pilots. From this plot, some categories can be easily identified such as the rolling takeoff as it is the only takeoff type where the vehicle has an indicated airspeed above 15kts before the liftoff point. The categories A and B can also be identified and differentiated from the plot. Figure 5 depicts the algorithm, working similarly to a decision tree, to classify each takeoff using the altitude and the ground speed. To differentiate normal takeoffs and maximum performance takeoffs, a support vector machine (SVM) model was implemented and trained as the takeoff profiles can appear similar on the altitude versus airspeed plot.

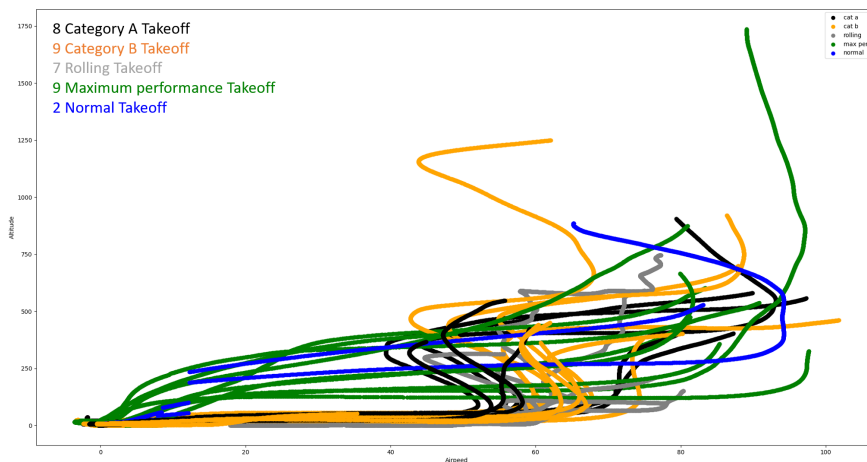


Fig. 4 Takeoff profiles of the different takeoff categories

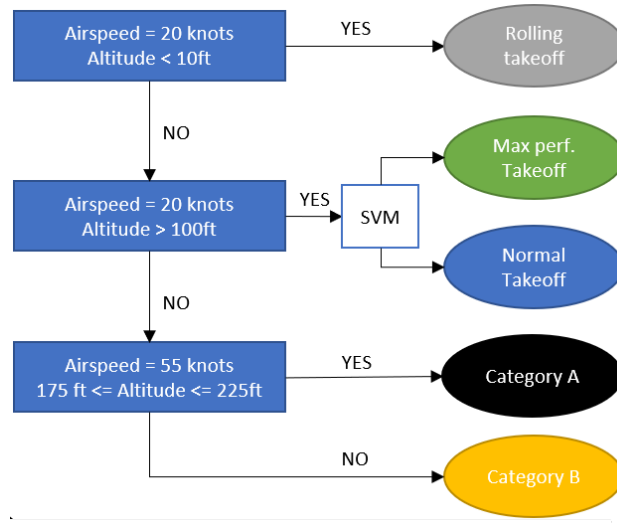


Fig. 5 Takeoff classification algorithm

C. Outlier detection

To detect anomalies in different takeoff categories, a sequential approach was proposed with the main steps are shown in Figure 1. Using a sampling of several takeoffs (already pre-processed by category) a data cleaning process starts. First, during the initial phase of the takeoff, there are turbulent effects that cause random behaviors on the airspeed and altitude curves. This type of data cannot be used as part of a training/test of a neural network and must be disregarded. Based on the end of the initial turbulent phase, an airspeed lower bound is set.

After that, depending on the takeoff category, the airspeed and altitude upper bound limits are set and the datasets are filtered using those limits as reference. At this moment, the dataset that does not present enough data points to fulfill these limits is disregarded.

Before selecting and preparing the training and test datasets to run the Neural Network, all of them must have the same length. In that case, the smaller dataset length is used as a standard and the lengths of the other datasets are reset starting from the upper limits. At that moment, the takeoff datasets are ready to run on the RNN-LSTM, so the training and test datasets are defined and used to run the Neural Network. The first step, before run the neural network is define the features to be used, once the takeoff raw data has more than 80 different features that could be used to improve the RNN performance. Due to the lack of real datasets, it is hard to evaluate the performance of the Neural Network and avoid problems, such as overfitting, to create a robust model. To address this issue, synthetic data was generated from the available datasets of each type of takeoff. The detail of using synthetic data will be explained in the following section. The goal is to create an altitude-by-airspeed curve to be used as a prediction model. This curve is a combination of two time series.

The prediction model of altitude by airspeed is created for each type of takeoff and this model is used to create a typical curve for each takeoff category and it will be used as a reference to a modified z-score outlier detection. These vectors are modeled to fit a multivariate Gaussian distribution using Maximum Likelihood Estimation. This last modelling step is finally used to detect the outliers; resulting in anomaly detection of FDM safety events.

1. Feature selection

The helicopter raw data used during the analyzes of this project has air speed and altitude, that are the main parameters used to create the helicopters takeoff curves, but these are only two of more than eighty features such as, horizontal flight path angle, roll acceleration, pitch acceleration, yaw acceleration, and others. Different features could be used to improve the neural network performance. Feature selection techniques offer significant benefits by enabling the identification and selection of the most crucial and highly-ranked features from a dataset.

Feature selection has been a widely researched topic in both methodology and practice for several decades. Theoretical principles suggest that feature selection methods can be based on statistics, information theory, manifold and rough set, and categorized based on various criteria.[15]

- i. Based on the utilized training data, feature selection methods can be classified as supervised, unsupervised, or semi-supervised models. A unified framework for all three models of feature selection has been proposed.
- ii. Feature selection methods can be categorized as filter, wrapper, or embedded models based on their relationship with learning methods.
- iii. Feature selection methods can be derived from correlation, Euclidean distance, consistency, dependence, and information measure based on the evaluation criterion.

The feature analysis was already a subject of previous studies performed by the FAA, in that case, to avoid miss the main objective of the current analysis with long discussions features importance, the feature selection used, based on the inputs available and the method efficiency, was a supervised type called Extremely Randomized Trees Classifier (Extra Trees Classifier), a type of ensemble learning technique which aggregates the results of multiple de-correlated decision trees collected in a “forest” to output it’s classification result. In concept, it is very similar to a Random Forest Classifier and only differs from it in the manner of construction of the decision trees in the forest. [15]

2. Recurrent Neural Network – Long Short-Term Memory

Neural Networks present connections only between adjacent layers and there is no connection between nodes in the same layer. All inputs and outputs are independent of one another. Dealing with problems that involve temporal relationships is challenging.

RNNs using loops in them could be a solution to this problem. With a standard RNN architecture, information can be transmitted from one step to the next step because the hidden units receive feedback from the previous state to the current state. It is possible to think of the unrolled structure as numerous copies of the same network. Keeping this in mind, one finds that the final output depends on both the input and the result of the hidden layer that came before.

The ability to learn long-term temporal correlations is constrained by the difficulties of using traditional RNN in long-range dependencies, the gradient either expands or vanishes during back propagation. Long-Short Term Model (LSTM) was proposed to solve this issue. At the core of the LSTM architecture is the concept of a memory cell that is capable of retaining its state over time, accompanied by non-linear gating units that manage the inflow and outflow of information from the cell [16]. A typical schematic of an LSTM is shown in Figure 6.

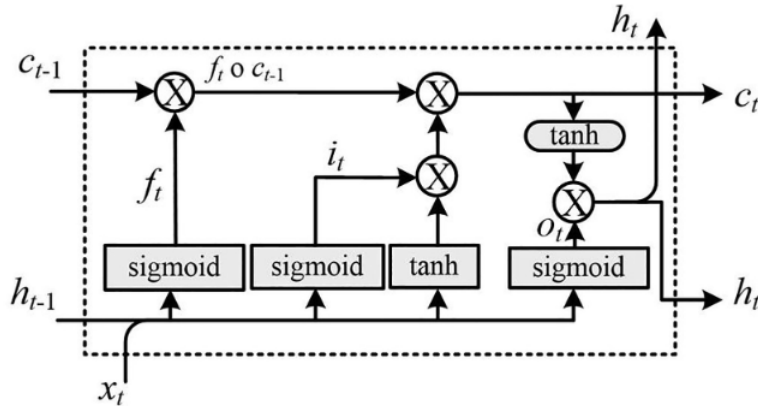


Fig. 6 Detailed Structure of LSTM Cell [16]

Considering x_t as the input and h_t as the output at time t . At this time t , the equation of gates, memory cell, input and output of the LSTM cell are shown below.

$$i_t = \sigma(W_i[x_t] + R_i[h_{t-1}] + b_i) \quad (1)$$

$$f_t = \sigma(W_f[x_t] + R_f[h_{t-1}] + b_f) \quad (2)$$

$$o_t = \sigma(W_o[x_t] + R_o[h_{t-1}] + b_o) \quad (3)$$

$$g_t = \tanh(W_x[x_t] + R_x[h_{t-1}] + b_x) \quad (4)$$

$$c_t = f_t c_{t-1} + i_t g_t \quad (5)$$

$$h_t = o_t \tanh(c_t) \quad (6)$$

Where i_t , f_t , o_t states input gate, forget gate and output gate respectively and c_t defines the memory cell to store the past state. The g_t and h_t states to input and output of the LSTM cell respectively and h_{t-1} states to the output of the previous LSTM cell, coefficients and weights W , R and b . The learning algorithm is a method that involves propagating the error backwards through a neural network, and updating the weights and biases in a way that minimizes the error with respect to the current training sample. This process is iteratively repeated to train the neural network. The weight W is updated using the following equation [17].

$$\Delta W = -\eta \frac{\partial E}{\partial W} \quad (7)$$

where η is the learning rate and E is the error. The partial derivatives of the factors c_t and s_t are used to calculate the weight by deriving the error with respect to the activation, and the activation in terms of the state.

$$\Delta W = -\eta \frac{\partial E}{\partial c_t} \frac{\partial c_t}{\partial s_t} \frac{\partial s_t}{\partial W} \quad (8)$$

Sigmoid and tanh are the activation functions to map the non-linearity. An improved alternative to the LSTM is change the activation function, part of a contemporary approach in deep learning systems involves replacing saturated activation functions such as sigmoid and tanh with non-saturated ones like ReLU. This offers advantages such as resolve the issue of "exploding/vanishing gradient", and speed up the convergence process.

So, using RNN with LSTM cells is a solution to prevent the long-term reliance problem and retain information for extended periods of time. The application of this neural network was to create a model for each takeoff type which were used as a reference pattern to identify takeoff outliers that could be failures over different sets of data collected. The takeoff models were created based on the air speed and altitude behavior during this phase of the flight.

3. Modified z-score

The modified z-score (z_m score) is utilized to measure how far a data sample is from the mean value of typical observation [18]. In this study, the typical observation is a profile standard or reference defined by the neural network model. The distance from takeoff (Y) and the model (Y_m) is measured using the Mean Squared Error (MSE). The z_m score is computed dividing the distance by the minimum sample standard deviation.

$$MSE = \frac{1}{N} \sum_{i=1}^N (Y_i - Y_{m_i})^2 \quad (9)$$

After compute the z-score for each takeoff dataset, in necessary define a threshold that will be used to identify the outliers over the sample available.

4. Threshold definition

Most widely used thresholding techniques are based on statistics like standard deviation around mean and median absolute deviation [19]. These statistics can be significantly biased, depending of the number of outliers in the sample when calculating these statistics. In cases of high number of outliers in the sample, other techniques, such as iterative threshold computations could be applied. Three threshold methods used were Standard Deviation (SD), Median Absolute Deviation (MAD), and Clever Standard Deviation (Clever SD). The mean and SD are the corresponding statistics of the z_m scores.

- Standard Deviation

$$T = mean + a \cdot SD \quad (10)$$

- Median Absolute Deviation

$$T = mean + a \cdot MAD \quad (11)$$

$$MAD = b \cdot median(|X - median(X)|) \quad (12)$$

- Clever Standard Deviation

Iterative approach using SD method, but the initial step is compute a maximum range value *Clever R_{max}* using the same Equation 10 with $a = 2.5$. Any score higher than R_{max} is removed and the new R_{max} is computed until no more scores are greater than this value.

The constants a and b could be defined by the user, the commonly values are $a = 3$ and $b = 1.4826$ [19].

V. Experiments and Results

In this section, we implemented the outlier detection method described in the previous section. The process began with dataset augmentation and feature selection. Next, RNN models were generated for each takeoff category, serving as representatives of the typical takeoff behavior. These RNN models were then used to compute the modified z-score for each takeoff available in the datasets. To evaluate the method and performance of various threshold methods for this type of data, we utilized three test cases. The results were assessed to determine the most effective threshold method for the given data.

A. Dataset augmentation and features selection

To prevent overfitting, machine learning applications are very dependent on a huge amount of training data. Unfortunately, there is usually not enough data available for time series analysis. In this case, data augmentation is a powerful method to increase the quantity and quality of the training data; a consideration that is essential for the effective use of machine/deep learning models.

To improve the models created by the RNN runs on the takeoff analysis, synthetic data was created based on the takeoff datasets available for each type of them. To generate the synthetic takeoffs noise was added to the takeoffs considered normal or baseline patterns for each takeoff type. The noise was added separately to each time series type (altitude and airspeed) used in this analysis, then recombined to create the synthetic takeoff curve. The synthetic data generated for each takeoff category is shown in Figure 7.

The minimum number of datasets to run a Neural Network is something difficult to define, considering that exists a large number of different factors that affect the results, such as the complexity of the problem, the type of input data, and the complexity of the learning algorithm, for instance. Based on the available takeoffs dataset, for each category, the curve patterns do not present much complexity, on the other hand, there are some significant variations when the altitude and airspeed begin to increase. In addition to it, even in the categories with more datasets available, like category B with nine takeoffs, the only number of datasets with all the minimum conditions to use as training datasets are five, due to the number of points available and the curve patterns. Therefore, based on the research of Hüsken et al. [20] on recurrent neural networks for time series classification the number of synthetic datasets created from each "training" takeoff was fifty and the total number of available and synthetic takeoff datasets are shown in Table 1.

Takeoff category	Original Takeoffs	"Training" Takeoffs	Synthetic Takeoffs
A	8	5	250
B	9	4	240
Rolling	7	4	280
Maximum Performance	9	4	240
Normal	8	5	250

Table 1 Number of takeoffs datasets and synthetic data

As mentioned on subsection IV.C, the Extra Trees Classifier method was applied to identify the four most important features [21]. For the airspeed, the other most important features to improve the neural network performance were barometric altitude, absolute altitude, climb or descent rate, and copilot radio altitude. For the altitude feature, the other most important features were true airspeed, cyclic position roll, pitch acceleration, and copilot radio altitude as shown in Figure 8 and Figure 9.

B. Results of takeoff models and outlier detection

The final datasets with the synthetic data were divided in 70% for training, 20% for test and 10% for validation. The total number of datasets for each takeoff category is in the range of 240 to 280 as shown in Table 1, which resulted in a total of 168 to 175 training datasets. The learning rate used was 0.001 and the number of neurons was 30 [22] using the ReLU activation function. After run the RNN-LSTM for all the categories, the training and validation loss results have similar patterns and presented good convergence as shown in Figure 10.

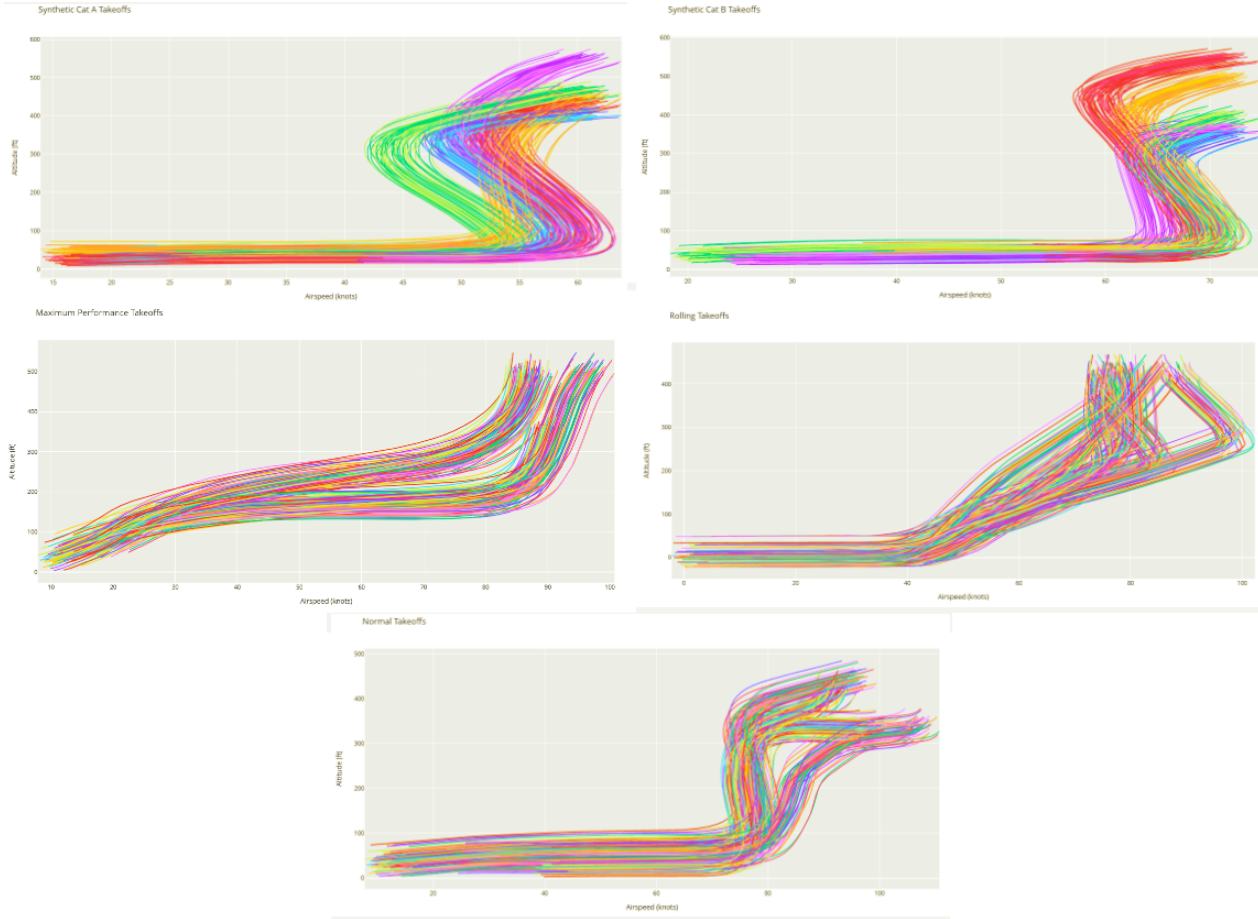


Fig. 7 Synthetic data generated for takeoff Cat A (first row, left), Cat B (first row, right), Max Performance (second row, left), Rolling Takeoff (second row, right), normal (third row)

The takeoff categories A and B original datasets have less variations of altitude and airspeed, so the neural network models show a good representation of the takeoff pattern. The normal takeoff original datasets presented more variations in airspeed. On the other hand, the maximum performance takeoff original datasets have more variations in the altitude. In both cases, the neural network model resulted in a flight path near to the average value of each parameter (altitude and airspeed) for these two categories. The rolling takeoff original datasets presented more variations in both parameters, but the model still captured these variations. The neural network models are shown as red bold curves in Figure 11.

After generate the models, the first outlier detection test has been done using the category B datasets available from simulations, two possible possible takeoff failure curves generated changing the helicopter takeoff trajectory to a descent one using two different altitudes based on the Helicopter Flying Handbook [13] takeoff common errors. The first one near to 210 ft (Takeoff 6) and the second one near to 80 ft (Takeoff 8) as shown in Figure 12

The category B takeoff model was used to compute the z_m based on the MSE of each original takeoff dataset computed point by point, starting from the last point available (final altitude) of each dataset and airspeed greater than 50 knots. The MAE and z_m of are shown in Table 2. The threshold computed using the SD method was equal to 7.58949 for $a = 0.5$ in equation 10, so the outliers takeoff are the number 6 and 8 (bold rows in the same Table). The takeoff number 8 presented the higher z_m score because the curve started to change with a lower altitude than the takeoff 6, which means that the takeoff number 8 has more "outliers points" than the other one.

The other two methods of threshold computation were used and the results are shown in Table 3. The datasets were the same used before with the same z_m score values. The equations and methods used were explained in the section IV.C. The MAD method presented the higher threshold value. The constants used in the equations 11 and 12 were $a = 0.5$ and $b = 1.4826$ [19], the drawback is that some takeoffs that could be an outlier may not be considered in different dataset combinations.

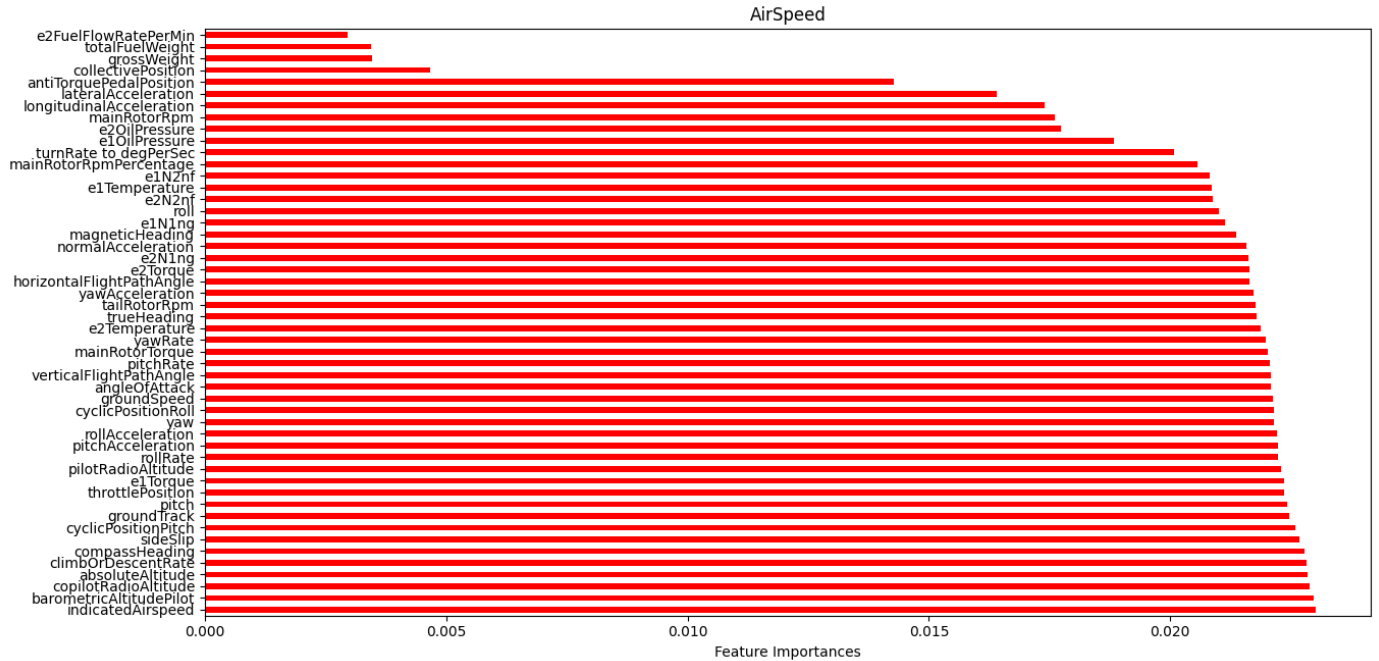


Fig. 8 Airspeed features importance

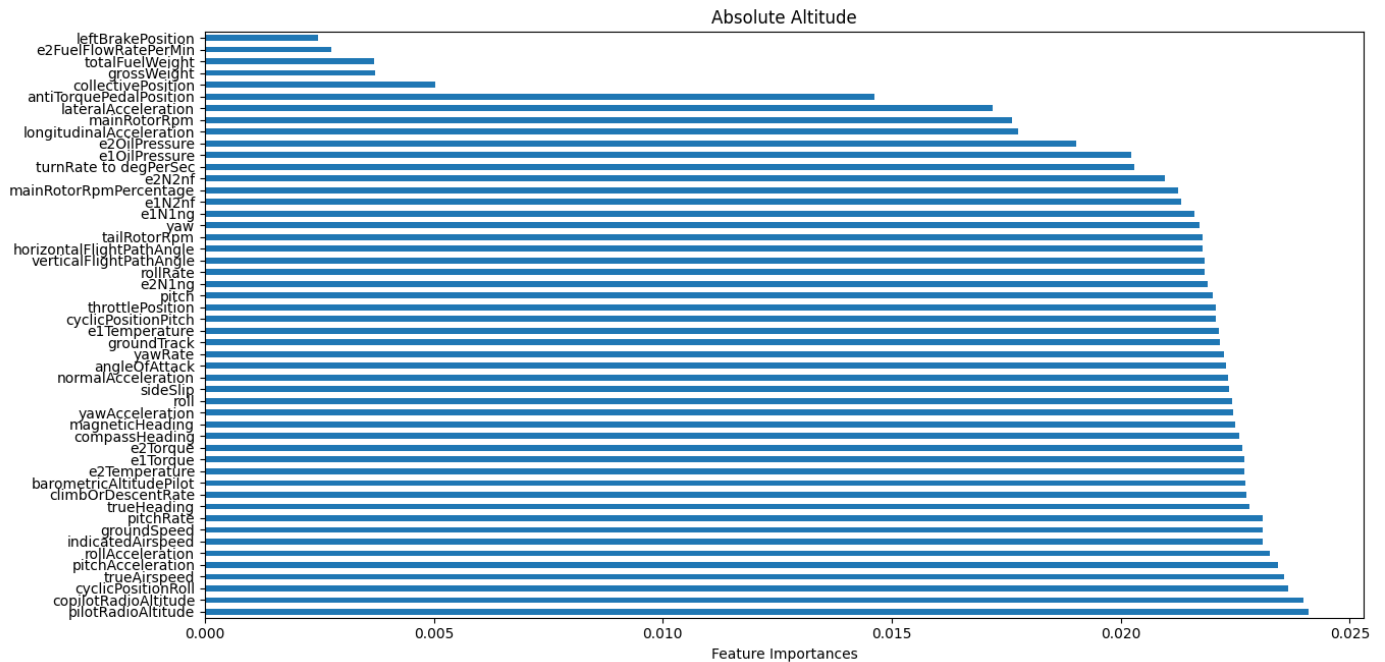


Fig. 9 Altitude features importance

The Clever SD method is the opposite, with a lower threshold, for instance, some non-outliers takeoff could be considered as outliers in different dataset combinations. This method is an iterative one, in that case, the first *Clever* $R_{max} = 12.92807$, so any z_m score greater than R_{max} should be removed, so the Takeoff 8 (z_m score = 14.87345) was removed. On the next iteration, the *Clever* $R_{max} = 7.80628$, this time the Takeoff 6 (z_m score = 8.88441) was removed. On the next iteration *Clever* $R_{max} = 3.94187$, in this case, the higher z_m score was the Takeoff 3

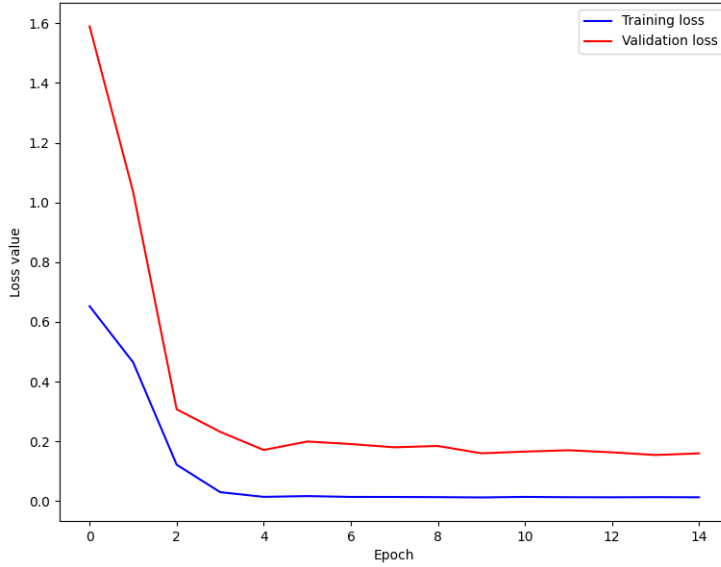


Fig. 10 Training and validation loss results

Takeoff number	MSE	z_m score
1	2.70093	2.25266
2	3.65313	4.62693
3	4.62693	3.8590
4	3.80186	3.17087
5	2.76740	2.30810
6	10.6523	8.88441
7	2.91704	2.43290
8	17.8332	14.87345

Table 2 MSE and z_m score results

($z_m = 3.8590$) which is lower than R_{max} . Therefore, there is no more takeoffs to remove and the takeoffs 6 and 8 were identified as outliers. The SD method could be considered as an intermediate alternative with respect to the other two.

Method	Threshold	Outliers detected (Takeoff number)
SD	7.58948	6 and 8
MAD	8.19349	6 and 8
Clever SD	5.06324	6 and 8

Table 3 Threshold results for different methods

After this initial analysis, a more generic case was assessed. The following case would represent a real application of the method, considering a larger sample, again for the takeoff category B. The synthetic data generated for the neural network augmentation was used again, now 200 takeoff were used and one outlier was added. In the first case, discussed before, even by inspection, the outliers takeoff 6 and 8 could be detected. In this next case, it is not an option as shown in the Figure 13 of the 201 takeoffs.

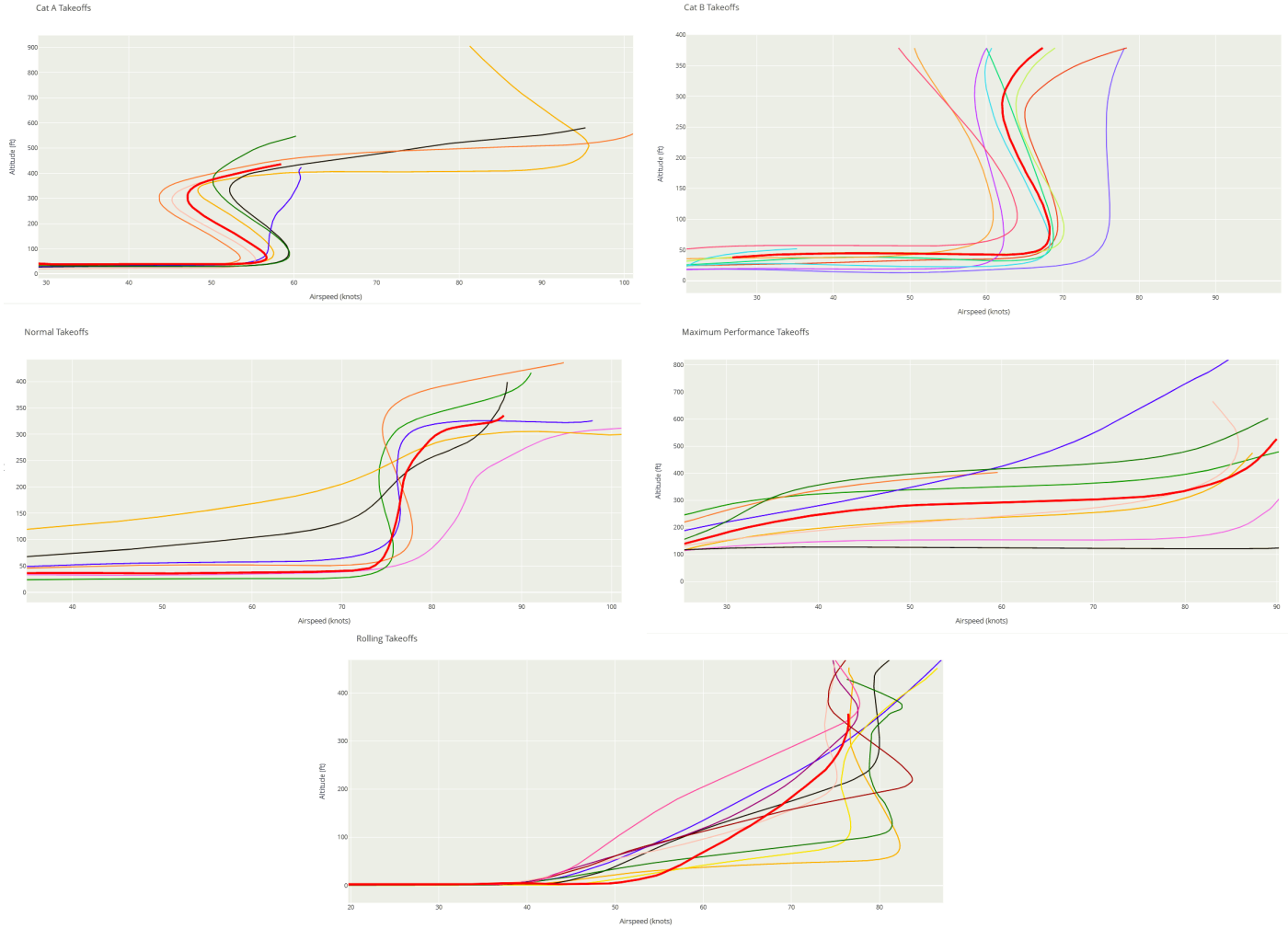


Fig. 11 Neural Network models in red for takeoff Cat A (first row, left), Cat B (first row, right), normal (second row, left), Max Performance (second row, right), and Rolling Takeoff (third row)

All the three methods were used to detect the outlier. The z_m scores for all the takeoffs were computed and they are shown in Figure 15. The threshold for each method was computed: SD: 18.00120; MAD: 18.6133; Clever SD: 20.39294 (first iteration). For the SD method takeoffs number 3 ($z_{msocre} = 18.50732$) and 101 ($z_{msocre} = 46.36451$) were detected as outliers. The method MAD detected only the takeoff 101 as an outlier and the Clever SD detected only the takeoff 101 as an outlier in the first iteration.

The next Clever iteration was implemented. So, the takeoff 101 was removed and the threshold was computed again. For the second iteration, $T_2 = 17.39212$, so takeoffs 3 and 24 (z_m score = 17.85965) were identified as outliers and removed for the next iteration. On the third iteration, the $T_3 = 16.70557$, as there were no z_m scores greater than T_3 , so the outliers detected were takeoffs 101, 3 and 24.

The summary of the three threshold methods are shown in the Table 4. Therefore, the MAD method was the only one without false positive detection, capable to identify the outlier takeoff 101 highlighted in the Figure 16. After that, the process was repeated, using the MAD threshold method to was used for all the takeoff categories. The outlier was detected in all the cases as shown in Figure 17.

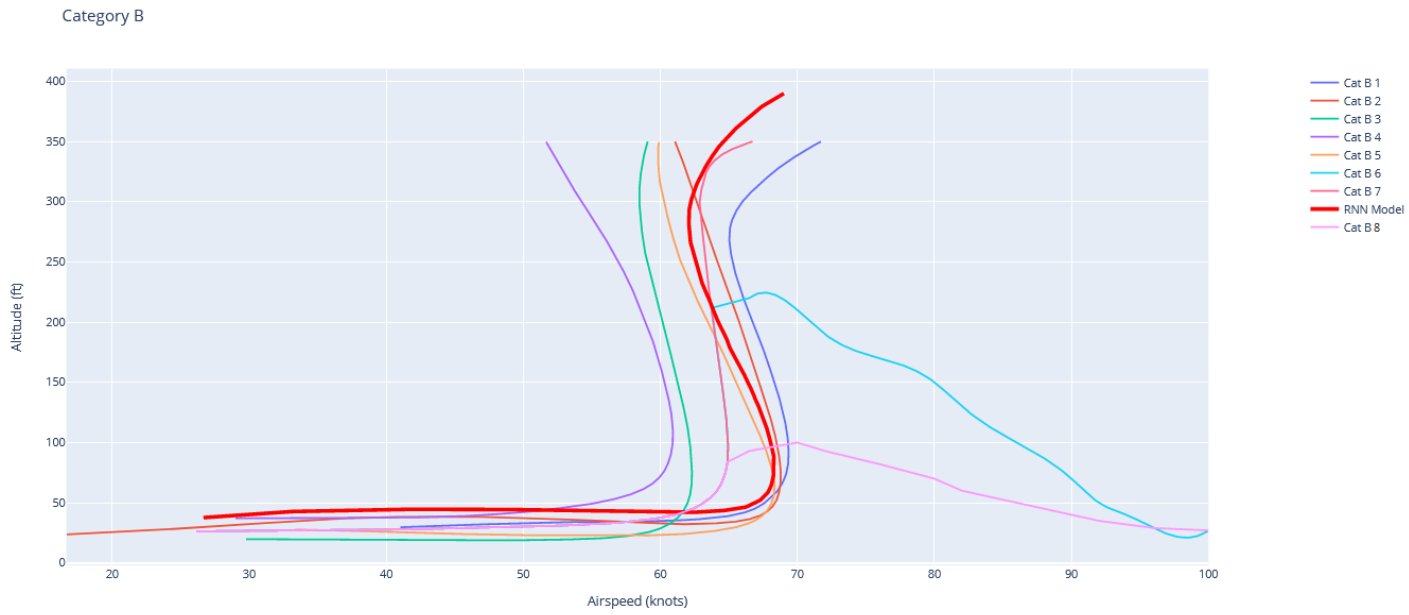


Fig. 12 Category B failure simulations

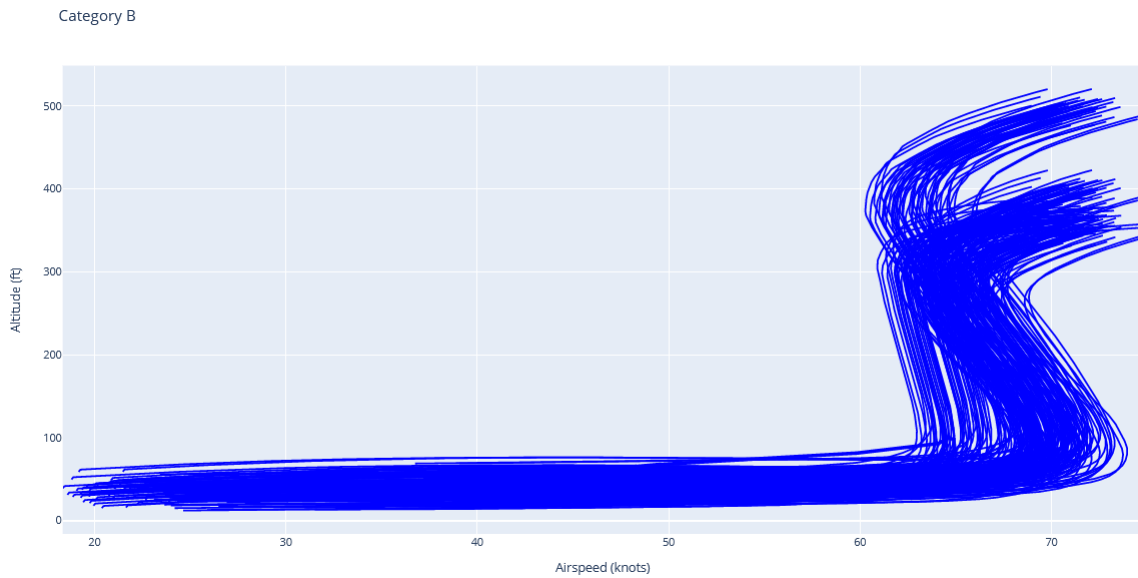


Fig. 13 Category B - 201 takeoffs

Method	Threshold	Outliers detected (Takeoff number)
SD	18.00120	03 and 101
MAD	18.61331	101
Clever SD	16.70557	3, 24 and 101

Table 4 Threshold results for different methods - case 2



Fig. 14 z_m scores for takeoff category B



Fig. 15 z_m scores for takeoff category B - Clever iteration

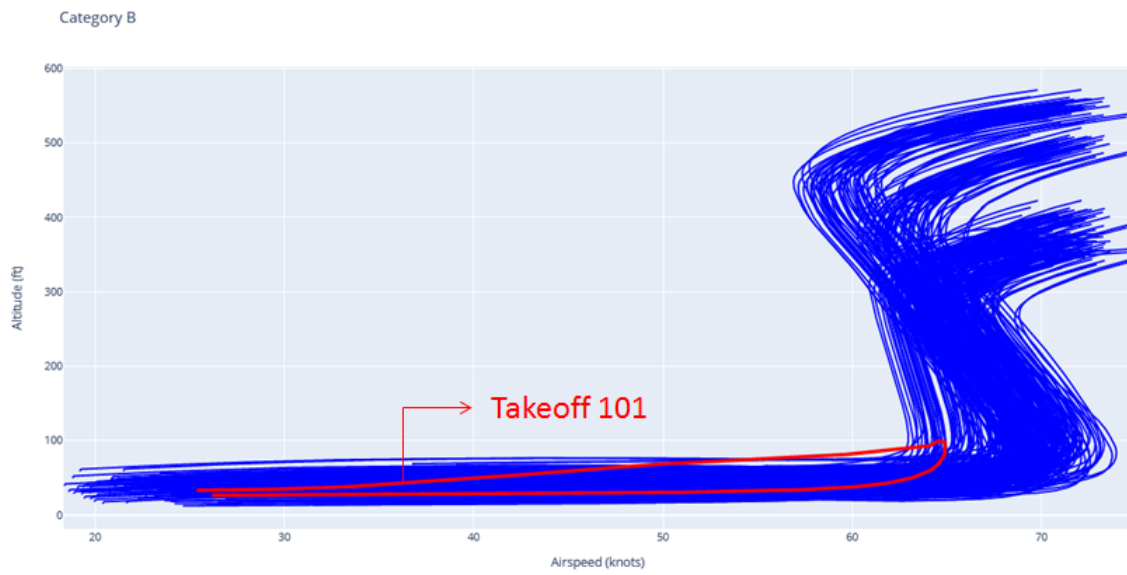


Fig. 16 Takeoff 101 highlighted

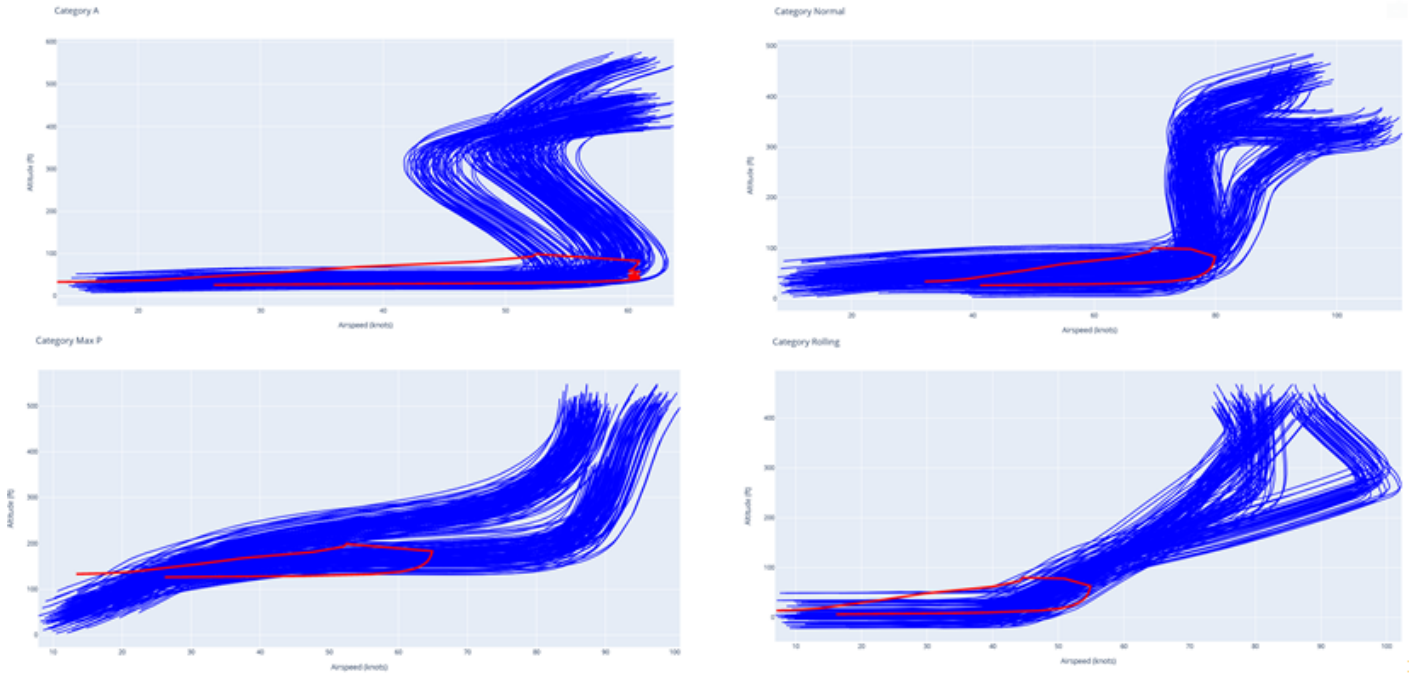


Fig. 17 MAD applied for all other takeoff categories

Even in cases like the rolling takeoff, where there are two "groups" of z_m scores as shown in Figure 18, the MAD method could detect only the takeoff 101 as an outlier.

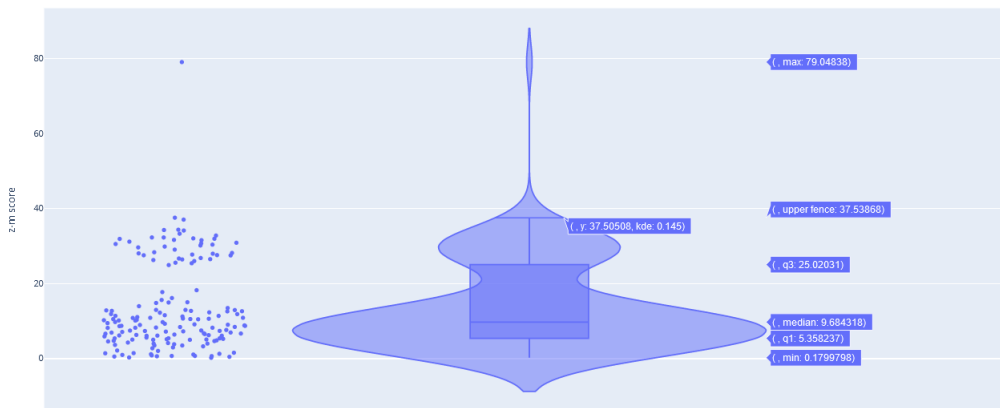


Fig. 18 Violin plot for rolling takeoff

The final case of study was based on the FAA pilot's suggestion of possible unusual takeoff situations that could be defined as a category A or category B. The pilots suggested analyze confined area takeoffs, in that case, the initial part of the takeoff (before increase the altitude) would be the same of a regular takeoff, however the maximum altitude is limited due to a presence of an obstacle, limiting the altitude of the takeoff which is constant for a certain period of time. The maximum altitude suggested was near to 150 ft or 200 ft with an airspeed between 56 and 100 knots. This scenario was modeled considering a category B takeoff.

Starting from the previous case, other four 'confined area takeoffs' were added to the dataset, again they are not easy to detect by inspection as shown in Figure 19. The modified z-score were computed using the airspeed as the reference feature. Based on the airspeed modified z-score, using the MAD threshold method, five outliers were detected. The

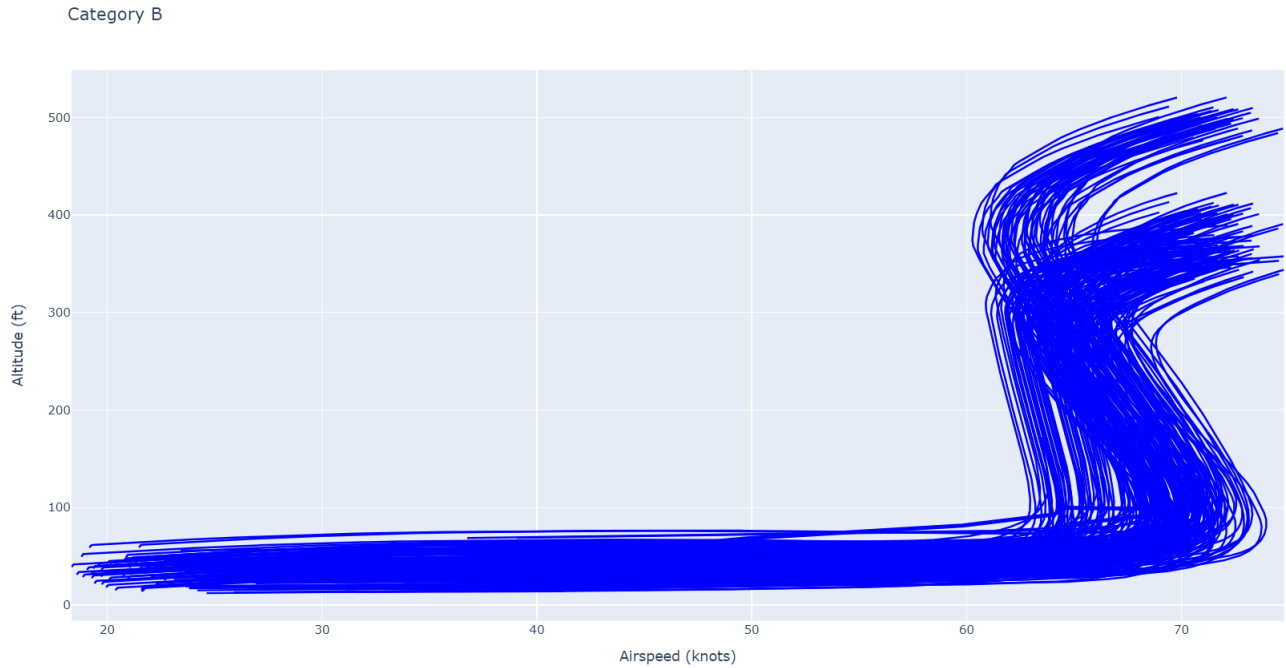


Fig. 19 205 category B takeoff dataset

MAD threshold, in this case, was 105.879807 and only two real outliers were detected (takeoffs number 77 and 117), the other outliers detected were false positives (takeoffs 3, 24, and 54), these results are shown in Figure 20 where the threshold is the red horizontal dotted line. The takeoff curve is based on the airspeed and altitude, in that case, the modified z-scores were computed again, but this time using the altitude as the reference feature. For this other test, the

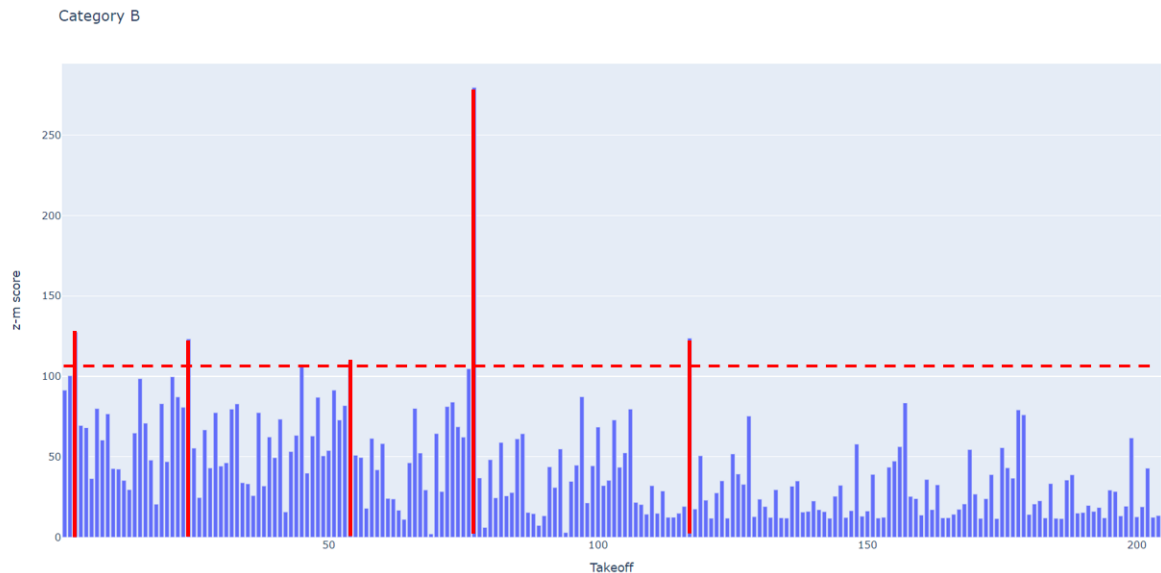


Fig. 20 Modified z-score using the airspeed

MAD threshold was 483.24352 and only the outliers takeoffs were detected (takeoffs 37, 73, 77, 117, and 179) as shown in Figure 21, where the threshold, again is the the red horizontal dotted line. This result is associated to the final airspeed of the outliers that are near to the final airspeed of the other takeoffs in this dataset. However, as the final altitude of the

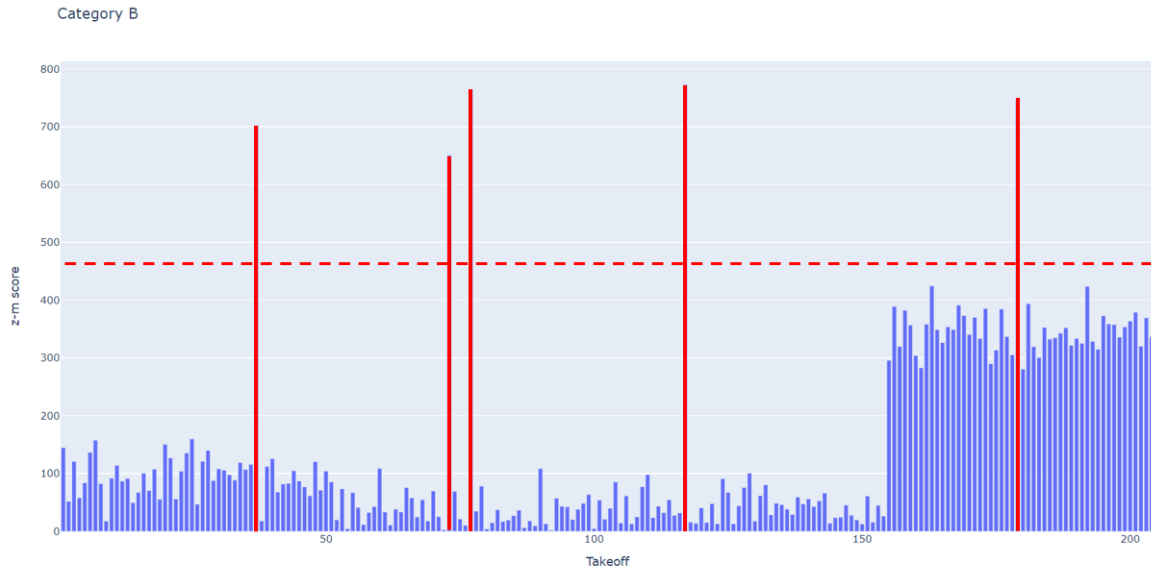


Fig. 21 Modified z-score using the altitude

outliers are significantly less than other takeoffs (Figure 22), they could be easily detected using the MAD threshold.

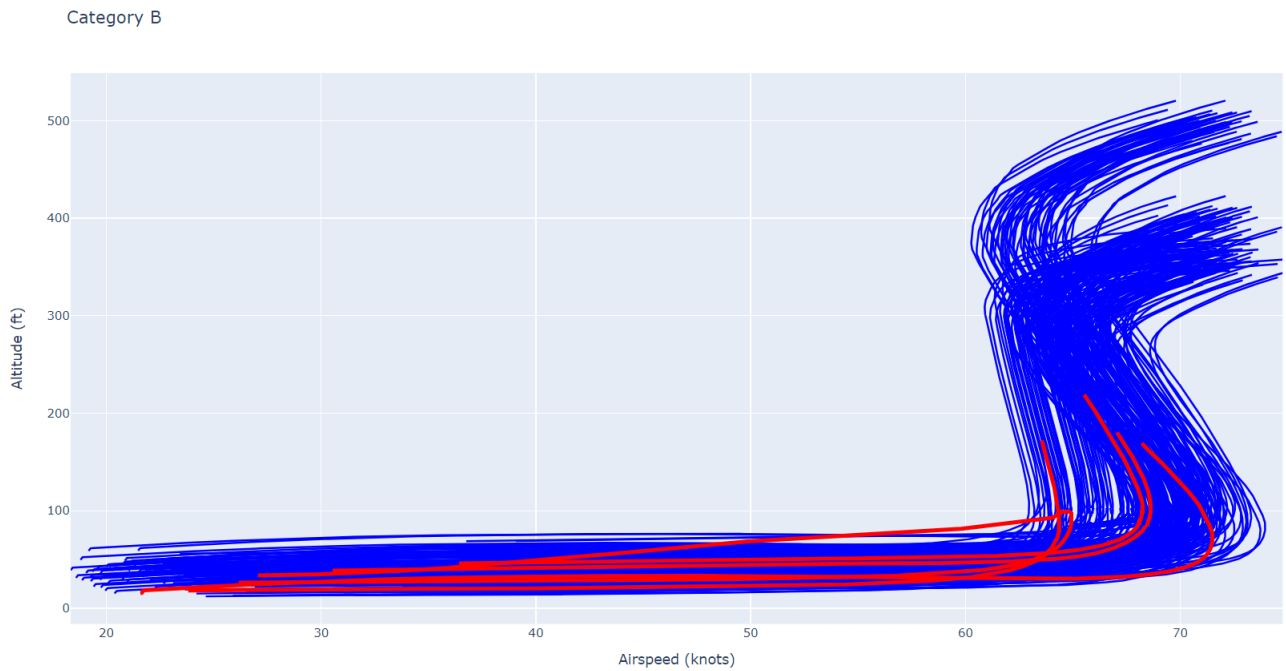


Fig. 22 Category B outliers identified

VI. Conclusions

The objective of this research is to develop a methodology for identifying anomalies during the helicopter takeoff phase. The approach comprises of multiple steps, starting with the takeoff segment identification, takeoff classification, create neural network models, compute z_m score values and define a threshold to identify the outliers. The critical step to properly identify the outliers is in the last step during the threshold definition, which depends of the statistical parameters of the datasets sample available, threshold constant definitions and the desired “filtering” criteria.

Different threshold methods were used to identify the outliers. Considering a small set of takeoff, the outliers were detected using the three methods proposed, however depending of the desired detection performance, each one presented different applications. To avoid high threshold values and "miss" outliers datasets, the Clever SD method is a good alternative. However, as shown in the test case 2, it could present false positives.

Using the outliers cases suggested by the pilots (case 3), using only the modified z-score with the respect to the airspeed, some outliers were not detected. However, using the altitude as the reference feature, all the outliers were detected without false positives. So, the outlier takeoff could present differences in airspeed and altitude combined (the outliers could be detected using only one feature analysis) or more differences in only one of the features (only the altitude or only the airspeed). Base on the results, the modified z-score and MAD threshold is a useful method to identify outliers in takeoff datasets. However, from the test case 3, is important to emphasize that the method must be applied to both features to avoid 'miss' outliers that do not present significant differences in one of the reference parameters in some cases.

There are several potential future works that could build upon our present research. Some of these are outlined below:

- Collect more takeoff data to run the Neural Networks and improve the model’s fidelity
- Test the methodology to different helicopter phases
- Explore other outlier detection techniques

Acknowledgments

This work is funded by the Federal Aviation Administration through PEGASAS (Partnership to Enhance General Aviation Safety, Accessibility and Sustainability), FAA Center of Excellence on General Aviation, Project No.2: Rotorcraft Aviation Safety Information Analysis and Sharing (ASIAS). Project No. 2: Rotorcraft ASIAS is a partnership between PEGASAS researchers at the Georgia Institute of Technology, the FAA and Helicopter Association International (HAI). The information presented in this paper and contained in this research does not constitute FAA Flight Standards or FAA Aircraft Certification policy.

References

- [1] USHST, “Monthly Safety Report,” https://www.ushst.org/Reports/Safety_Report20200904.pdf, 2020.
- [2] Payan, A. P., Gavrilovski, A., Jimenez, H., and Mavris, D. N., “Improvement of rotorcraft safety metrics using performance models and data integration,” *Journal of Aerospace Information Systems*, Vol. 14, No. 1, 2017, pp. 26–39.
- [3] NTSB, “Enhance Recorder Use to Enhance Safety,” *NTSB 2017-2018 Most Wanted List of Transportation Safety Improvements*, 2017.
- [4] NTSB, “Aircraft accident report,” *Washington, D.C. : U.S. Govt.*, 2018.
- [5] Chin, H.-J., Payan, A. P., Johnson, C. C., and Mavris, D. N., “Anomaly detection in initial climb segments for helicopter operations,” *Vertical Flight Society’s 77th Annual Forum*, 2021.
- [6] Meng, C., Jiang, X. S., Wei, X. M., and Wei, T., “A Time convolutional network based outlier detection for multidimensional time series in cyber-physical-social systems,” *IEEE Access*, Vol. 8, 2020, pp. 74933–74942.
- [7] Su, Y., Zhao, Y., Sun, M., Zhang, S., Wen, X., Zhang, Y., Liu, X., Liu, X., Tang, J., Wu, W., et al., “Detecting outlier machine instances through gaussian mixture variational autoencoder with one dimensional CNN,” *IEEE Transactions on Computers*, Vol. 71, No. 4, 2021, pp. 892–905.
- [8] Kieu, T., Yang, B., and Jensen, C. S., “Outlier detection for multidimensional time series using deep neural networks,” *2018 19th IEEE International Conference on Mobile Data Management (MDM)*, IEEE, 2018, pp. 125–134.

- [9] Zhong, J., Zhang, Y., Wang, J., Luo, C., and Miao, Q., "Unmanned aerial vehicle flight data anomaly detection and recovery prediction based on spatio-temporal correlation," *IEEE Transactions on Reliability*, Vol. 71, No. 1, 2021, pp. 457–468.
- [10] Qin, K., Wang, Q., Lu, B., Sun, H., and Shu, P., "Flight Anomaly Detection via a Deep Hybrid Model," *Aerospace*, Vol. 9, No. 6, 2022, p. 329.
- [11] Rey, M., Aloise, D., Soumis, F., and Pieugueu, R., "A data-driven model for safety risk identification from flight data analysis," *Transportation Engineering*, Vol. 5, 2021, p. 100087.
- [12] Ning, S., Sun, J., Liu, C., and Yi, Y., "Applications of deep learning in big data analytics for aircraft complex system anomaly detection," *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, Vol. 235, No. 5, 2021, pp. 923–940.
- [13] Handbook, H. F., "FAA-H-8083-21A," *United States Department of Transportation, Federal Aviation Administration, Airman Testing Branch*, 2019.
- [14] Chin, H.-J., "Retrospective and exploratory analyses for enhancing the safety of rotorcraft operations," Ph.D. thesis, Georgia Institute of Technology, 2021.
- [15] Jie Cai, S. W. S. Y., Jiawei Luo, "Feature selection in machine learning: A new perspective," *Neurocomputing*, 2017.
- [16] K. Vijayaprabakaran, K. S., "Towards activation function search for long short-term model network: A differential evolution based approach," *Journal of King Saud University - Computer and Information Sciences*, 2020.
- [17] Ralf C. Staudemeyer, E. R. M., "Understanding LSTM – a tutorial into Long Short-Term Memory Recurrent Neural Networks," 2019.
- [18] Seem, J. E., "Using intelligent data analysis to detect abnormal energy consumption in buildings," *Energy and Buildings*, 2006.
- [19] Jiawei Yang, P. F., Susanto Rahardja, "Outlier Detection: How to Threshold Outlier Scores?" *AIIICC '19: Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing*, 2019.
- [20] Hüsken, M., and Stagge, P., "Recurrent neural networks for time series classification," *Neurocomputing*, Vol. 50, 2003, pp. 223–235.
- [21] Chulhee Lee, d. D. A. L., "Decision Boundary Feature Extraction for Neural Networks," *IEEE TRANSACTIONS ON NEURAL NETWORKS*, Vol. 8, No. 1, 1997.
- [22] Pankaj Malhotra¹, G. S. P. A., Lovekesh Vig, "Long Short Term Memory Networks for Anomaly Detection in Time Series," *ESANN 2015 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, 2015.