Flight Data Monitoring (FDM) Unknown Hazards detection during Approach Phase using Clustering Techniques and AutoEncoders

Antonio Fernández, Darío Martínez
Pablo Hernández and Samuel Cristóbal
Innaxis Research Institute
Madrid, Spain
Email: {af,dm,ph,sc}@innaxis.org

Florian Schwaiger
Institute of Flight System Dynamics
Technische Universität München
Germany
Email: f.schwaiger@tum.de

José María Nuñez
and José Manuel Ruiz
Iberia airlines
Madrid, Spain
Email: {jmnunez,jmruizn}@iberia.es

***Abstract*—Airlines safety departments analyse aircraft data recorded on-board (FDM) to inspect safety occurrences. This activity relies on human experts to create a rule-based system that detects known safety issues, based on whether a small set of parameters exceed some predefined set of thresholds. However, rare events are the hardest to manually detect, as patterns are not often recognised at glance. Experts agree that both approach and take-off procedures are more prone to experience a safety incident. In this paper we performed descriptive and predictive analyses to detect anomalies during approach phase for runway 25R in LEBL airport. From a descriptive point of view, clustering techniques aids to find patterns and correlations within data, and also to identify clusters of similar observations. Moreover, this clusters might reveal certain points as rare events that are isolated from the rest of the observations. Predictive analytics, and more concisely deep learning ANN and AutoEncoders, can be used to detect this abnormal events. The methodology relies on learning how ”normal” observations looks like, since they usually are the majority of the cases. Afterwards, if we process an abnormal flight, the model will return a high reconstruction error because of the deviation from the training data. This shows how the predictive methodology could be applied as an extremely useful forensic tool for safety experts and FDM analysts.**

***Keywords*—Anomaly detection, hazard identification, safety, clustering, deep learning, LSTM, AutoEncoders, HDBSCAN**

I. Introduction

Flight Data Monitoring (FDM) is an activity carried out by airlines primarily for monitoring and improving the safety and operation of their aircrafts [1]. The data recorded by the flight data recorder on-board an aircraft is downloaded and analysed through various tools and techniques, with the ultimate objective of using that analysis to improve civil aviation operations, establishing maintenance schedules, training pilots and modifying operational procedures amongst others, without compromising safety. The benefits of analysing FDM are often highlighted by the safety departments, that analyse occurrences. This activity relies on human experts to create a rule-based system that detects known safety issues, based on whether a small set of parameters exceed some predefined set of thresholds. This unveils the necessity of introducing predictive analysis methodologies to automate the detection of events [2]. In particular, rare events are the hardest to manually detect, as patterns are not often recognised by an analyst. Furthermore, the identification anomalous and rare flights is a challenging task in proactive safety management systems.

From the data science point of view, anomaly detection is a machine learning discipline that involves outlier detection, deviation detection, or novelty detection. Abnormal events detection and cause-effect analysis in aviation represent a challenging field that involves the mining of multi-dimensional heterogeneous time series data, the lack of time-step-wise annotation of events, and the challenge of using scalable mining tools to perform analysis over a large number of events [3]. Machine learning (ML) models that work on datasets under imbalanced constraints (such as a very low count of anomalies) are limited and may not be able to learn patterns that explain the anomalies correctly [4]. This means that the cause-effect analysis might be incomplete or not well supported. Also, the study might need to consider some interesting anomalies that are not ”labelled” in the historical data but need to be investigated. In this context, we need to shift the approach to make the model less dependent on labeled data and consider a semi-supervised machine learning methodology [5].

Semi-Supervised algorithms can be used to train a model to learn the behaviour of the ”normal approaches”, which compound the majority of the dataset. We understand as a ”normal approach” those observations which features values are bounded within the rest of the distribution. By applying this methodology, we only require labels during the pre-processing phase as we need to obtain a dataset only containing approaches operated under normal conditions. After this “non-anomalous” dataset has been obtained, it can be used to train an unsupervised machine learning model, i.e. without the need for labels and learning the latent representation of the data.

With enough data, deep learning algorithms ensemble different layers to build a neural network model. In particular, AutoEncoders are a special type of neural network architectures that exploit the presented semi-supervised methodology [6]. To classify the abnormal flights, several approaches might be followed. For instance, the reconstruction error can be calculated by comparing the characteristics and patterns of the abnormal

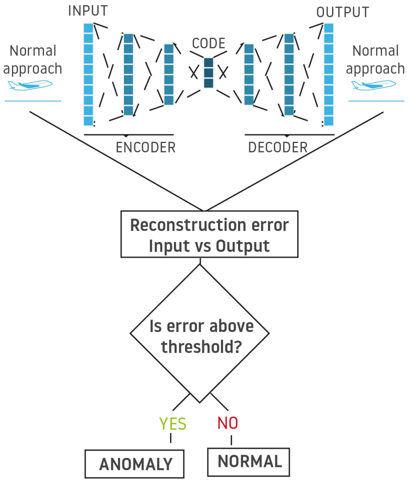


Figure 1: Anomaly detection using AutoEncoders, measuring the reconstruction error between input and output

flight with learned low level representation of a normal flight and measuring the loss difference. Flights over an empirically set threshold will be considered ”abnormal”. However, one of the core features supported by AutoEncoders is the possibility of studying apparently normal flights classified as abnormal, leading to new types of safety hazards.

Using supervised methodologies (e.g. binary classification based on LSTMs) is expected to under-perform due to the imbalanced nature of the training data, i.e. the lack of anomalous approaches. Because of this, a semi-supervised architecture makes more sense.

For this study we have analysed a set of approaches landing on LEBL, more concisely on runway 25R. First, performing a descriptive analysis mainly focused on applying clustering techniques to detect similarities between the observations, quantifying an ”outlier score” per sample. Second, to perform a predictive analysis over the outlier samples, training an autoencoder considering only those observations with a low outlier score, and predicting anomalous approaches by measuring the reconstruction error between input and output. The reconstruction error will be high for those flights that differ a lot from the regular procedure. The output of the analysis can be used to automatically flag anomalous flights to be later analysed by safety experts. The output of recommendations for the crew or the safety analysts are out of the scope of the research.

Therefore, the main objectives of the proposed research are two:

1) *Descriptive analysis based on an unsupervised clustering.*
2) *Predictive analysis based on a semi-supervised AutoEncoders neural network.*

Section II presents a literature review of the state of the art of ML techniques applied anomaly detection and flight data monitoring (FDM) analysis. Section III defines the problem, the context of the proposed scenario and the scope of the descriptive and predictive analyses. Section IV describes the studied dataset; then explains how to prepare the data to perform the clustering and train the AutoEncoder. In Section V, we present the methodology with a descriptive data analysis of some arrivals, with a detailed analysis of the context and patterns detected in the outliers. Furthermore it also present a predictive analysis to detect anomalous approaches. Section VI puts the results in perspective, showing how the predictive methodology could be applied as a forensic data analysis tool for safety experts. Section VII finally concludes with a discussion about the analysis performed and the results obtained.

II. State of the Art

Anomaly detection is one of the main topics in data science research, though many challenges still await solution. Machine learning (ML) provides a brand new toolkit for tackling aviation problems. And, in particular, it has been the increasing trend of using ML to analyse flight data efficiently, as it offers the most important insights into the operations of an aircraft. The difficulties for applying predictive analytics to FDM data have been leveraged in the past, remarking the significance of a reliable features selection pipeline [7]. Furthermore, the process of automatically select attributes from a given dataset by the learning algorithms itself is well known capability in ML ensemble methods [8].

The capabilities of automatic features selection algorithms have not reached full potential for FDM data, only two major works tried to tackle the issue [9] [10]. In a recent attempt, Bro et al. (2017) [11] proposed a ANN-based methodology for predicting go-arounds, using more than 2.000 hours of FDM data of training flights. Since the work done in features extraction for FDM is very limited, we believe that more recent deep learning methodologies such as AutoEncoders can release the full potential of FDM predictive analytics.

Regarding anomaly detection research in aviation, NASA/Ames Research Center applied Multiple Kernel for Anomaly Detection (MKAD) for anomalous event detection within American terminal manoeuvring areas, combining both continuous and static features [12]. Due to the complexity of training a kernel-based machine learning algorithm, the study was only performed over a dataset of a few thousand arrivals. In recent work, they have improved the computational time by applying deep learning method, specifically extreme learning machines based in AutoEncoders and embeddings [13]. The NASA/Ames Research Center has also defined and experimented with methodologies for improving the efficiency of the investigation of anomalies [14]. They aim to distinguish between operationally significant anomalies and uninteresting statistical anomalies based on a weak-supervised classifier. AutoEncoders have also already been used to find breakpoints in time series [15] and to predict realistic transitions in sector configurations [16].

Xavier Olive et al. [17] presented at the SIDs 2018 a very interesting approach with AutoEncoders and combining speech and ADS-B data, to detect anomalies in controller’s actions. Mainly focusing in deviations of the flight trajectory from flight plan. This paper is of the highest relevance given that the techniques applied and problem context share similarities with this paper. However, this paper will be focused in analysing FDM data and the anomalies detected will often be safety related.

III. Problem assessment

In this paper, we present a diagnostic analytics problem, with the ultimate outcome of finding out dependencies and identifying patterns that help better define the causes for anomalies in the approach phase or even find new anomalies undetected by experts.

Normally, airlines safety departments manually inspect individually all their flights looking at concrete operations that might hide safety implications behind. However, this procedure could be quite tedious since the majority of the observations usually behave normally. Machine learning might empower this labor learning how normal procedures are operated, and automatically discriminate the outliers to be further analyzed by safety experts. Furthermore, this could reveal new possible safety concerns not previously detected by the airlines.

In order to narrow down the problem, we decided to focus on the approach and landing phases rather the whole flight trace, since departure and arrival procedures are more likely to experience a safety incident. Additionally, we constrained the scope of the research to a particular procedure, so we selected FDM approaches landing on runway 25R of LEBL airport. This aims to decrease the noise introduced to our scenario and extract particularized conclusions that could be extrapolated to a wider number of airports or runways in the future.

The research process has been carried out by performing a descriptive and predictive analysis. The descriptive analysis focuses on inspecting the data, and creating clusters of flights that landed under similar circumstances. Moreover, we will research about rare events detection and examine specific flights that differ from the rest of the distribution analysing the causes, which might go from external influence factors such as weather or traffic congestion, to wrongly calibrated or broken sensors, and even issues during the FDM decoding.

The predictive analysis will take the set of outliers identified in the descriptive analyses, and will attempt to automatically classify these abnormal approaches, training an AutoEncoder and measuring the loss obtained as output. A well-trained AutoEncoder should be able to predict correctly normal approaches as they will have similar patterns following the same distribution. The reconstruction error will be small for these cases; nevertheless, it will increase if we introduce a rare-event as an input. By catching these high errors and establishing an empirical threshold, we can perform a binary classification of normal and abnormal flights.

IV. Data

The main data source used to perform the analyses has been FDM data. However, other data sources such as METAR, for weather information during the approach, and the final flight plans have been used to provide an external context to the dynamic information captured by aircraft sensors (e.g. ETA, ATOT, origin aircraft, STAR, etc...).

The dataset is composed by 35.000 approach operations in LEBL 25R. This means that our dataset contains successful and failed (e.g. go-arounds, touch-and-go, ...) landing attempts. FDM data is known for presenting a very high variable dimensionality with more than 150 different variables stored as time-series, with a resolution of up to 8 samples per second. Given that iterative clustering techniques tend to be memory intensive we decided to reduce the number of features by down-sampling the temporal series. The new sampling rate was selected by ensuring an observation every 0.5NM between the touchdown (TD) point and 12 NM from TD. In this way, we have a complete overview for the approach context with one point every 30 seconds, from the beginning of the landing procedure to the touchdown in the runway.

An scalable infrastructure is mandatory to run the complete pipeline described in Figure 2. We have used DataBeacon [18] platform to process the 35.000 flights, which is based on Amazon Web Services stack. The cluster used was an EMR cluster composed by 1 master node and 5 workers, deployed on *m5.4xlarge* instances with a CPU of 16 GB and 64 GiB for memory.

V. Methodology

A. Feature engineering

The feature engineering process, both for the descriptive and predictive analytics, have been computed in an unique multi-step data preparation pipeline including data cleaning, de-identification of sensitive information, data merging of the three datasets (FDM, METAR and Flight Plans) and time-points extraction (12NM to TD every 0.5NM). The complete pipeline executed from decoded FDM files to extract the input dataset is described in Figure 2. Increasing the granularity of the sampling rate would lead to less loss of information, but more computational capabilities to train the models would be required, so we finally sampled at 0.5NM which provides a good trade-off between throughput and granularity of data. The features have been grouped in several categories which are summarized in Table I

In a final remark, each landing attempt/approach is dealt independently. The destination airport for subsequent attempts

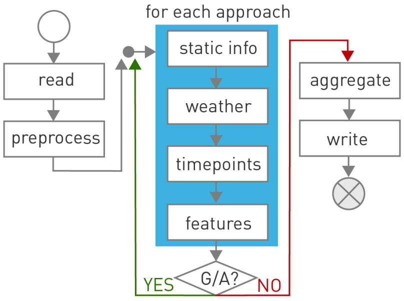


Figure 2: Data preparation pipeline for FDM data. G/A means go-around, and if it happens, then a new approach attempt is created.

TABLE I: Datasets and features

Group	Features	Data Sources
Operation dynamics	Pitch, roll and heading positions and rates. Angle of attack. Vertical descent rate. Barometric altitude. Radio altitude	FDM
Aircraft energy	Air speed. Ground speed. Energy level. Aircraft mass.	FDM
Adverse weather	Static pressure. Static temperature. Relative humidity. Dew point Air density. Wind direction. Wind speed. Wind variation. Ground visibility. 1st Cloud layers height and opacity.	METAR (LEBL) FDM (aircraft sensors)
Aircraft configuration	Flaps orientation	FDM
Crew coordination	Autopilot status. Pilot flying	FDM
Pilot awareness	Current time, Distance travelled, Total Time Flown	FDM
Flight static information	UTC time. Origin Airport. Destination Airport. Call sign. Aircraft type. Tail number. Year. Week. Runway Occupancy Time. Time at threshold. STAR. SID. Time exit. Runway. Runway exit	Flight Plan FDM

might be different, so information about the airport (e.g. runway exit) cannot be propagated across the whole flight. The same applies to weather reports taken from METAR that can refresh between multiple attempts. After running the pipeline we obtained a total of 825 features per approach attempt to feed our models.

B. Descriptive analysis: Clustering

In this section we will explore our input dataset looking at variables distribution and potential correlations existent between features. The dimensions of the input dataset are 35.000 samples and 825 features. It is composed by multiple landing attempts with static and dynamic features. As it can be noticed, our data has a high number of columns (825), which complicates the data visualisation. In order to deal with this problem, dimensionality reduction techniques were applied to the dataset to better represent the features. This data transformation pipeline can be considered as a sophisticated feature engineering process, i.e the output will be the input of the clustering algorithm.

The selected dimensionality reduction algorithm is the t-distributed Stochastic Neighbor Embedding (tSNE) technique [19]. tSNE is a probabilistic algorithm that minimizes the divergence between pairwise similarities of the input objects and their corresponding low-dimensional representation respectively. In other words, it inspects the input statistical properties and manages to represent this data using less dimensions by matching both distributions in the best way. It has been proved that it is well suited for the visualisation of high-dimensional datasets [19]. The memory and computational requirements needed to run this algorithm are quite high since tSNE scales quadratically in the number of objects contained in the input. Therefore, learning the 2D representation of the data is very slow for datasets larger than a few thousands of input observations.

In Figure 3 the 2D representation of the data is visualized. By only looking at the picture, it is easy to visually recognize groups of flights that are very close to each other. In the other hand, it can be noticed how some points are completely isolated from the rest of the distribution, or in the middle between two clusters. These points can be considered as outliers for the given distribution. Nevertheless even if clusters are recognizable at glance, we performed a clustering algorithm to group all the samples in regardless categories.

HDBSCAN is a clustering algorithm developed by Campello, Moulavi, and Sander [20]. It extends DBSCAN by converting it into a hierarchical clustering algorithm, and then using a technique to extract a flat clustering based in the stability of clusters. Results after applying HDBSCAN algorithm to tSNE representation of the distribution is described in Figure 4, where it can be observed how the model is able to determine 9 different clusters.

HDBSCAN algorithm enables to flag certain points as noise, avoiding to set a specific cluster for these samples. Most of these noisy samples are placed between two different groups, or isolated away from any identified cluster. In addition to this feature, HDBSCAN supports the GLOSH outlier detection

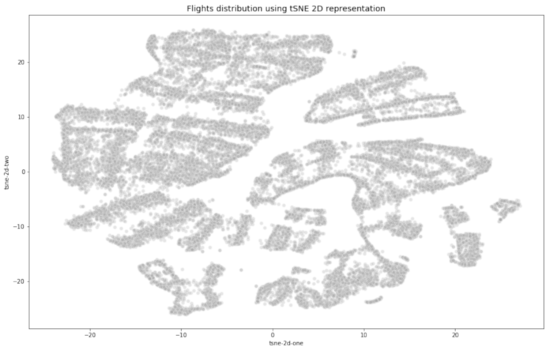


Figure 3: Two-dimensional representation of the input dataset using tSNE probabilistic algorithm.

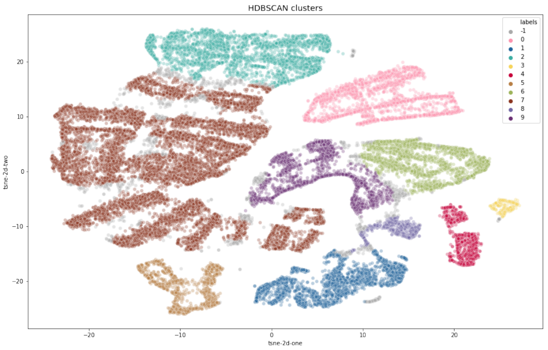


Figure 4: Clustering using HDBSCAN algorithm over tSNE representation.

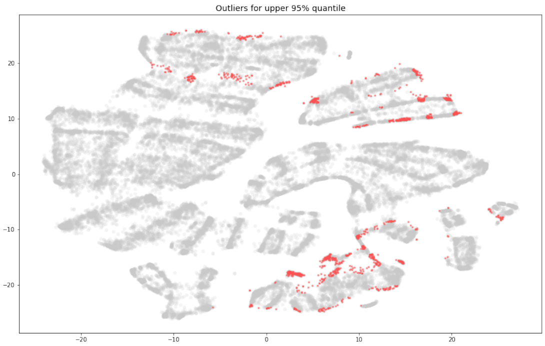


Figure 5: Two-dimensional representation for the outlier scores obtained from GLOSH algorithm in HDBSCAN clusters. The red dots are extreme outliers above 95% quantile.

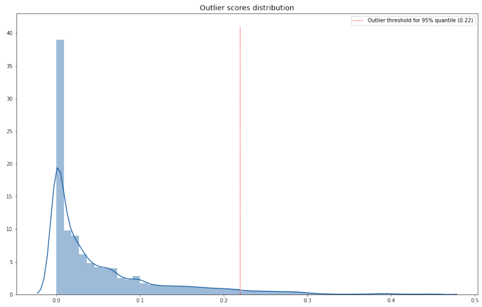


Figure 6: Histogram of outlier scores obtained from GLOSH algorithm. Threshold set for 95% quantile of the distribution. All points above the threshold are extreme outliers.

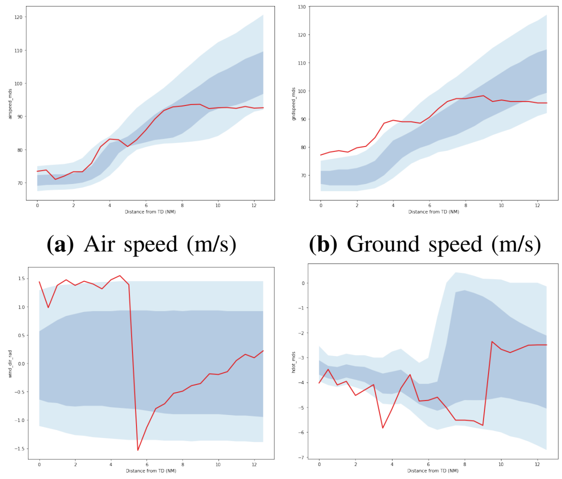
algorithm, and it allows to combine it with the clustering algorithm output. The GLOSH outlier detection algorithm is also related with outlier detection algorithms such as Local Outlier Factor (LOF), where anomalies are detected by measuring the local deviation of an observation with respect to its neighbours. It is a fast and flexible outlier detection system that implements the detection of ”local outliers”. Local outliers detection implies that the algorithm can detect outliers within a local region or cluster, that doesn’t need to be global outliers.

The algorithm outputs an outlier score for each sample of the dataset. Then a threshold is set for classifying outliers: being 0 a normal point and 1 an outlier sample. By calculating the 95% quantile, we can extract those points that present the highest outlier score. The scores distribution obtained after running GLOSH outlier detection algorithm over the HDBSCAN cluster is shown in Figure 5. In the figure, the dotted red line represents the threshold for the most extreme outliers. After plotting the detected extreme points in our tSNE distribution, location for anomalous points can be observed in Figure 6.

We have detected a total of 1.750 outliers that exceed the 95% quantile of the distribution. After exploring in depth some of these flights, we found out several potential flights that landed in abnormal conditions, even having sensors that were wrongly calibrated or experienced issues during the decoding process. To show different types of detected anomalies, we selected two examples that were tagged as outliers by the algorithm. In order to better understand the causes, we will analyse the features evolution from a FDM analyst point of view. This type of analysis helps us to understand what flights are considered abnormal and why, just by inspecting the FDM time-series values.

1) Outlier A - High ground speed justified by a strong tailwind: In *Outlier A*, upon further inspections, we can find two metrics that stand out as atypical. The first one is the ground speed. This flight presents a significantly high ground speed, around 150 knots, during the final stages of the approach (0NM-2NM from the TD).

On the contrast, the airspeed (CAS) is not also unusually high, approximately 140 knots which is inside what could be consider normal operation. The second metric that stands out in this flight is the rate of descent. The flight presents a sharp increase in the descent rate at about 2NM to 4NM from the threshold. At this distance the aircraft is approximately at 1000 feet AGL (3 Degree glide path) which is the usually the threshold by which an aircraft needs to be stabilized before continuing the approach. Most common Unstable Approach (UA) criteria applied a descent rate above 1000 feet-per-minute is considered an upper limit. The flight here reaches the 1150 feet-per-minute well above this threshold. Taking into account both metrics we can try to make assumptions of the causes for these deviations.



(a) Air speed (m/s) **(b)** Ground speed (m/s)

(c) Wind direction (rad) **(d)** Vertical descent (m/s)

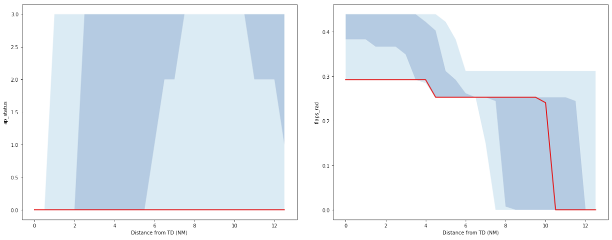
Figure 7: *Outlier A* features compared with the rest of the distribution percentiles. Horizontal axis represents the distance, from TD point to 12 NM from TD.

The best suited explanation is that the aircraft was suffering heavy tailwinds. The recorder wind speed at the 2NM-4NM distance was about 11 knots. A tailwind of such force could explain why the aircraft had an unusual high groundspeed while the airspeed was not especially high (the wind and aircraft trajectory are in the same direction). This hypothetical tailwind will also prompt an increase in the descent rate as the ground speed is increased. This exemplifies how the algorithm is able to detect anomalous flights caused by hazardous situations (Unstable approach) and how a safety officer can forensic analysis these results a produce useful insights. Metrics evolution over time are represented in Figure 7

2) Outlier B - Late Flaps deployment: This outlier has three metrics that stand out from the norm. The first one is the Auto-Pilot (AP) indicator. During this flight the AP is disengaged well before the final phases of the approach (as far as 12NM / 4000 feet). This is interesting because although it does not represent a hazard, as pilot can disengage the autopilot when safety can be assured, this is not normal procedure and most pilots maintain the AP engage until about 1500 feet or 4NM from threshold.

The metric causing the anomaly is the flaps position. In this particular flight, flaps are correctly deployed at about 100 feet but they are not fully deployed. Similar to the previous metric, this by itself does not represent a hazard but is rare as fully deployed flaps help control the aircraft at low speed and produces draft helping the aircraft to slow down to adequate landing speed. Finally, during the final approach phase the flight has to pronounce increases in the descent rate. Neither of the both increases in the descent rates surpass the 1000 feet-per-minute threshold but two sudden increases (<150%) in the decent rate could be a symptom of an unstable approach. These two increases could be caused by the flaps not being fully deployed, the aircraft speed decreases and there is not enough lift generated by the wings.

In contrast with *Outlier A*, this one has now special hazardous situation but had some metrics and performance that was out of the ordinary. This presents another possible benefit of these types of algorithms. It can help detect flights that without surpassing any defined threshold can be labelled as anomalous and help detect unknown hazards or risk behaviors in an unprecedented level. Metrics evolution over time are represented in Figure 8



(a) AP status **(b)** Flaps (rad)

Figure 8: *Outlier B* features compared with the rest of the distribution percentiles. Horizontal axis represents the distance, from touchdown point to 12 NM from TD.

C. Predictive analysis: AutoEncoder

The starting point for the predictive analysis are the outlier scores calculated in the descriptive analysis. These scores will be used to discern between abnormal and normal approaches. The AutoEncoder will be trained only using normal approaches to learn the behaviour of a non-anomalous approach. Once trained, the AutoEncoder compares each input flight with the learnt approach behaviour, outputting a reconstruction error score.

The objectives for the predictive analysis are:

- Design and implement an Artificial Neural Network (ANN) using an AutoEncoder architecture, composed by multiple LSTM layers to deal with the temporal series.
- Train and test the AutoEncoder, so it is tuned to recreate normal samples minimizing the loss, and in the other hand returning high reconstruction error when an outlier is processed.

As we commented before, we will design an AutoEncoder based on LSTM layers. LSTM is a type of recurrent neural network (RNN), very useful to extract patterns from sequential or time-series data. These kinds of models are capable of automatically extracting features from past events and LSTMs are specifically known for their ability to extract both long and short term features. LSTM is a bit more demanding than other models referring to data preparation. The input data to an LSTM model is a 3D array, with shape of $(n\_samples, n\_timesteps, n\_features)$. Hence the $n\_samples$ refers to the number of observations fed into the LSTM AutoEncoder, $n\_timesteps$ or look-back which describes the time window (past data) needed by the LSTM. Finally $n\_features$ represents the amount of columns selected as potential features for the scenario.

As detailed in Figure 1, AutoEncoders are composed by three main elements: encoder, code and decoder. We have designed an AutoEncoder with 2 layers for the encoding part, one single layer for the code and another 2 layers for the decoder. Therefore the neural network is composed by a total of 5 LSTM layers. For the activation functions we used ReLU function due to it popularity since it avoids and rectifies the vanishing gradient problem. The AutoEncoder receives a complete landing attempt as an input in a (1,825) vector format, encode it and decode it to finally return it minimizing the loss.

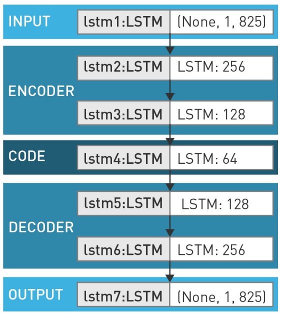


Figure 9: ANN layers for the LSTM AutoEncoder, showing the encoder, code and decoder layer sizes.

To train the auto-encoder, we have followed these steps:

- Divide the dataset in two parts, negatively labeled (normal approaches) and positively labeled (outlier approaches)
- Ignore the anomalies and train the auto-encoder only with negatively labeled data. Afterwards we will test the model using positively labeled data, to assess if model filter anomalies properly.

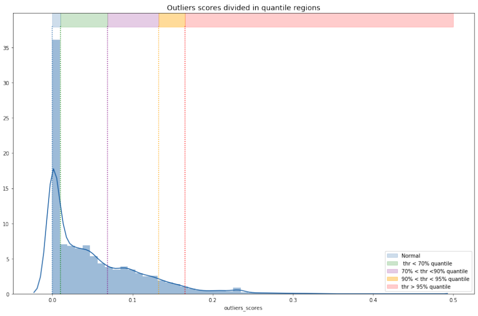


Figure 10: Histogram of outlier scores obtained from GLOSH algorithm. The red line represents the threshold for 95% quantile of the distribution. Those points above the threshold are extreme outliers.

The process of filtering which approaches are normal or not relies on analising the outlier scores obtained as descriptive results. If we check the outlier score distribution in Figure 5, most of the samples have a score around zero. We established four thresholds to categorize the ”normality” for samples with a score greater than zero. These thresholds are based on the distribution quantiles:

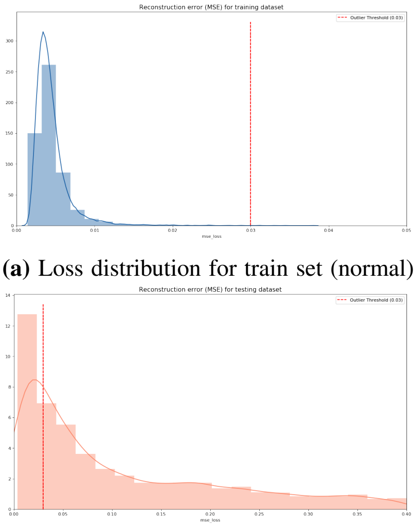
- Threshold 1 (THR1): near zero
- Threshold 2 (THR2): 70% quantile
- Threshold 3 (THR3): 90% quantile
- Threshold 4 (THR4): 95% quantile

These four regions are represented using different colors in Figure 10. This way we can measure how abnormal a sample is from those having a zero outlier score. For the input dataset composed by 35.000 flights, the distribution for each region is based on the anomaly severity:

- Normal ($outlier\_score$ = 0) - 12.011 flights
- Very low (THR1$<outlier_score<$THR2) - 12.489 flights
- Low (THR2$<outlier_score<$THR3) - 7.000 flights
- Medium (THR3$<outlier_score<$THR4) - 1.750 flights
- High ($outlier_score>$THR4) - 1.750 flights

With this outlier distribution we will label negatively only those having an outlier score equals to zero, and positively all the flights with an outlier score greater than 0, grouping them by how far from ”normal” behaviour they are. Thus, our training set has 12.011 flights using a 10% as a validation set, and our testing set has 22.989 flights, formed by anomalies of several degrees.

Once the AutoEncoder has been trained and validated, we should set an empirical threshold to classify normal from abnormal observations. To establish this threshold, we calculate the mean squared error (MSE) for the samples of the training dataset, to establish an optimal limit that contains as much normal samples as possible below the threshold. The Figure 11a represents the MSE distribution for the training set. The majority of the training samples return a mean error of 0.005, with some outliers that almost raise a loss of 0.04. Based on the error distribution, and the precision/recall curve, where we want to maximize the recall without loosing too much precision, we set the reconstruction error threshold at **0.03**



(a) Loss distribution for train set (normal)

(b) Loss distribution for test set (abnormal)

Figure 11: MSE histogram for training and testing sets. Reconstruction error threshold = 0.03

Afterwards, once the reconstruction error has been fixed is time to analyse how well the AutoEncoder filters the anomalies contained in the testing dataset.

VI. Results

To test the performance of the predictive model, a selected batch of anomalous data was inputted. We measured the accuracy of classifying correctly an anomaly, finding out that the AutoEncoder is able to classify more than 74% of the anomalies, predicting correctly almost all the Medium and High severity anomalies (Figure 12).

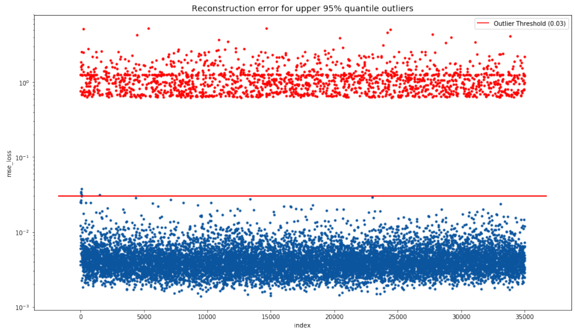


Figure 12: Reconstruction error for normal flights and outliers with a score higher than 95% quantile.

However, some flaws were detected. For example, the model struggles to classify samples with low outlier score. This is expected because low score samples are very similar to the ”regular” approaches that we used to train the model. Further research can be made to try to tackle this problem by following several methodologies. For example, adjusting more precisely the decision threshold, training with more samples, including additional meaningful features or using an even deeper ANN.

The loss distribution is represented in Figure 11b, where we can appreciate that most samples are located above the threshold. Furthermore, if we filter out the low to medium severity anomalies, and only consider the most severe ones filtering the upper 95% quantile, the AutoEncoder perfectly recognizes normal form abnormal observations. This is presented in Figure 12.

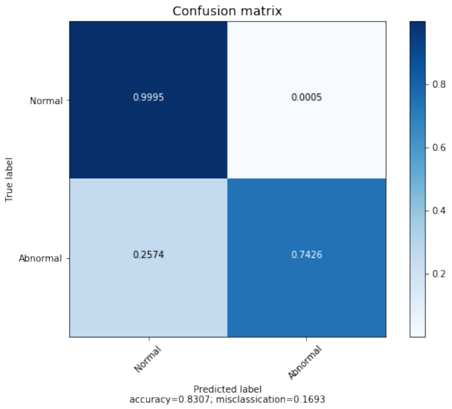


Figure 13: Confusion matrix for normal and abnormal (all severities) flights

The confusion matrix (Figure 13) exemplifies even further the main properties of the predictive model. The algorithm is very efficient at detecting very anomalous flights, but fails to set a distinguishable boundary between a normal flight and a flight presenting low severity hazard.

VII. Discussion

Given the promising results obtained for the 95% percentile anomalies, the algorithm provides an extremely useful tool for FDM and Safety analysts. An implementation could be an automatic FDM data labelling system. At the end of a day of operations, when FDM data is retrieved by the safety department, the system could flag anomalous flights for inspection. This can enable a tool for analysing and flagging large amount of FDM data. Not just for forensic analysis but also for aircraft maintenance, enabling inter-operability with existing predictive maintenance tools.

Another implementation of the methodology could be a real-time monitoring tool, which is able to ”quantify the risk” of the approach performed. This means that, by using the trained AutoEncoder, the tool could give an outlier score to a given flight. This information then could be used to warn airport crew, ATCOs, the airline or the crew.

Overall, the semi-supervised methodology has been successful. By combining two very powerful machine learning algorithm (HDBSCAN clustering and AutoEncoders), we were able to solve a very complex problem both from the data science and the aviation safety perspectives. The main benefits of using a semi-supervised architecture is to not only to automate and speed-up the detection of known events (e.g. late flaps deployment) but also to support the detection of unknown hazards and rare events.

The first part of the research has proven that, when using the adequate dimensionality reduction techniques, the automatic filtering of outliers is feasible. The second part of the research proved once again the efficiency of AutoEncoder architectures when working with time-series in aviation.

Acknowledgment

We want to thank the whole Safeclouds.eu consortium for providing the data sources and operational knowledge necessary to perform this research. With special mentions to TUM's institute of flight system dynamics for providing the algorithm for decoding and preparing FDM data. And Iberia airlines safety department for providing the operational expertise necessary to validate the results.

References

[1] R. Fernandes. An analysis of the potential benefits to airlines of flight data monitoring programs (msc). 2002.

[2] Sameer Jasra, Jason Gauci, Alan Muscat, Gianluca Valentino, David Zammit-Mangion, and Robert Camilleri. Literature review of machine learning techniques to analyse flight data. 2018.

[3] Vijay Manikandan Janakiraman. Explaining aviation safety incidents using deep temporal multiple instance learning, 2017.

[4] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM Comput. Surv.*, 41, 07 2009.

[5] Olivier Chapelle, Bernhard Schölkopf, and Alexander Zien. Semi-supervised learning. 2006.

[6] Pierre Baldi. Autoencoders, unsupervised learning, and deep architectures. In Isabelle Guyon, Gideon Dror, Vincent Lemaire, Graham Taylor, and Daniel Silver, editors, *Proceedings of ICML Workshop on Unsupervised and Transfer Learning*, volume 27 of *Proceedings of Machine Learning Research*, pages 37–49, Bellevue, Washington, USA, 02 Jul 2012. PMLR.

[7] Marius Kloft, Ulf Brefeld, Patrick Düssel, Christian Gehl, and Pavel Laskov. Automatic feature selection for anomaly detection. pages 71–76, 01 2008.

[8] Sergios Theodoridis and Konstantinos Koutroumbas. *Pattern Recognition, Fourth Edition*. Academic Press, Inc., Orlando, FL, USA, 4th edition, 2008.

[9] Bryan Matthews, Santanu Das, Kanishka Bhaduri, Kamalika Das, Rodney Martin, and Nikunj Oza. Discovering anomalous aviation safety events using scalable data mining algorithms. *Journal of Aerospace Information Systems*, 11:482–482, 07 2014.

[10] Lishuan Li and R. John Hansman. Anomaly detection in airline routine operations using flight data recorder data. *Report No. ICAT-2013-4, MIT International Center for Air Transportation (ICAT)*, 2013.

[11] John Bro. Fdm machine learning: An investigation into the utility of neural networks as a predictive analytic tool for go around decision making. *Journal of Applied Sciences and Arts: Vol. 1 : Iss. 3 , Article 3.*, 2017.

[12] Santanu Das, Bryan Matthews, Ashok Srivastava, and Nikunj Oza. Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study. pages 47–56, 07 2010.

[13] Vijay Manikandan Janakiraman and David Nielsen. Anomaly detection in aviation data using extreme learning machines. pages 1993–2000, 07 2016.

[14] J. Castle, J. Stutz, and D. McIntosh. Automatic discovery of anomalies reported in aerospace systems health and safety documents. 05 2007.

[15] Wei-Han Lee, Jorge Ortiz, Bongjun Ko, and Ruby Lee. Time series segmentation through automatic feature learning, 2018.

[16] Thomas Dubot. Predicting sector configuration transitions with autoencoder-based anomaly detection. 06 2018.

[17] Xavier Olive, Jeremy Grignard, Thomas Dubot, and Julie Saint-Lot. Detecting controllers' actions in past mode s data by autoencoder-based anomaly detection. 11 2018.

[18] DataBeacon. Remain in control of your data while optimizing your operations through artificial intelligence - databeacon.aero, 2019.

[19] Laurens van der Maaten and Geoffrey E. Hinton. Visualizing data using t-sne. 2008.

[20] Ricardo J. G. B. Campello, Davoud Moulavi, and Joerg Sander. Density-based clustering based on hierarchical density estimates. In Jian Pei, Vincent S. Tseng, Longbing Cao, Hiroshi Motoda, and Guandong Xu, editors, *Advances in Knowledge Discovery and Data Mining*, pages 160–172, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.