Journal of Air Transport Management 110 (2023) 102409

Contents lists available at ScienceDirect

Journal of Air Transport Management

journal homepage: www.elsevier.com/locate/jairtraman

A data analytics framework for anomaly detection in flight operations

Lucas Coelho e Silva [*], Mayara Condé Rocha Murça

Aeronautics Institute of Technology, Pca Marechal Eduardo Gomes, 50, São José dos Campos, São Paulo, 12228-615, Brazil

ARTICLE INFO

Keywords:
Anomaly detection
Machine learning
Flight data analysis
Aviation safety

ABSTRACT

In the air transport system, there has been a continuous effort to develop policies, tools, and methodologies that increase and standardize safety levels across the entire commercial aviation market, while also enhancing operational efficiency. Furthermore, there is a current focus on proactive approaches for aviation performance management. Within this context, data mining initiatives such as anomaly detection have become more prominent. Dealing with anomalies is a natural step for achieving the goals regarding operational safety and efficiency, as anomalies are often related to hazardous and inefficient operations. In this work, we propose a systematic flight data analytics framework for anomaly detection in flight operations in order to provide a comprehensive and reusable pipeline for model building, application, and explanation. The solution is designed to be applicable to both online and offline regimes and at multiple scales, while also building on domain-expert analysis. We demonstrate the framework applicability in two scenarios of routine flight operations monitoring considering both airline and air traffic management perspectives. In the first one, the framework is applied to aircraft performance data within an unsupervised learning setting with a density-based clustering approach for anomaly detection in landing operations at Minneapolis–Saint Paul International Airport (KMSP). The results are compared with those obtained with exceedance-based methods used in the current practice, revealing the detection of operationally significant anomalies beyond the benchmark. In the second case study, we apply the framework on flight tracking data within a supervised learning setting with the development of an autoencoder classifier for offline anomaly detection in terminal airspace arrival operations at Sao Paulo/Guarulhos International Airport (SBGR). Additionally, supervised learning models are developed for anomaly explanation. The autoencoder classifier was able to detect operationally significant anomalies, while the explanatory models provided novel insights about contributing factors to the anomalies identified. For instance, we learned that anomalous arrival trajectories are more likely to be associated with landing operations on runway 27 under wind scenarios, with an increase in the odds ratio of 62% and 58% for tailwinds and headwinds, respectively. In addition, we also observed a positive association between anomalies and wind gusts situations.

1. Introduction

With approximately 38 million scheduled commercial flights across 48,500 routes in 2019, the global aviation system is outstandingly complex (IHLG, 2019). Within the aviation system lies the flight operations subsystem, which deals with the day-to-day operations of flights. There are two principal stakeholders in the flight operations subsystem: airlines/aircraft operators and Air Traffic Management (ATM).

The utmost priority of flight operations is flight safety, be it from the airline or the ATM system point of view. Fatal accident rates for commercial flights have been continuously decreasing, making aviation the safest mode of commercial transport (Oster et al., 2013). However, current safety levels are not uniform across the world nor between each commercial aviation segment. There is thus a continuous effort to develop tools, policies, and methodologies that increase and standardize safety levels across the market. This is reinforced by the institutional emphasis and focus placed on flight safety. ICAO (International Civil Aviation Organization) launched the Global Aviation Safety Plan (GASP), which aims to achieve and subsequently maintain the zero-fatality level in commercial operations starting in 2030 (ICAO, 2019). The European Commission published in its vision for aviation the goal of reaching one accident or less per ten million commercial flights - a reduction of 80% compared to 2000 levels (European Commission, 2011).

One of the next challenges regarding safety is the change from an incident-based reactive approach to one which integrates a more proactive system-based predictive approach (Oster et al., 2013). The

[*] Corresponding author.
Linkedin: lucascoelho (L. Coelho e Silva).
E-mail address: coelho@ita.br (L. Coelho e Silva).

https://doi.org/10.1016/j.jairtraman.2023.102409
Received 16 November 2021; Received in revised form 7 April 2023; Accepted 7 April 2023
Available online 24 April 2023

proactive approach aims at identifying a hazard before its consequences happen, either fully or partially, whereas the traditional, reactive one, involves the identification of the hazard after its consequence happens (ICAO, 2013).

In addition to safety, the aviation ecosystem has also focused on increasing efficiency. Modernization programs such as FAA's Next Generation Air Transportation System (NextGen) in the U.S. FAA (2019), Eurocontrol's Single European Sky ATM Research (SESAR) in Europe (SESAR Joint Undertaking, 2019), and DECEA's SIRIUS in Brazil (Departamento de Controle Do Espaço Aéreo (DECEA), 2012) aim at increasing operational efficiency while guaranteeing safety levels with new technologies and procedures.

In parallel, aviation systems generate more and more data. From the ATM standpoint, modern and more widespread surveillance equipment such as Automatic Dependent Surveillance - Broadcast (ADS-B) provides detailed flight tracking information. From the aircraft operator perspective, better-equipped aircraft with modern Flight Data Recorders (e.g., QAR and DFDR) are capable of registering more parameters and at higher sampling rates.

Within this context of abundant data and focus on increasing safety levels via more proactive approaches while enhancing operational efficiency with the development of new evidence-based policies and practices, data mining initiatives for understanding and predicting aviation system behavior have become more prominent. One of these initiatives, which has drawn attention over the past years, is anomaly detection. Anomalies are patterns on data inconsistent with the expected behavior (Chandola et al., 2009). They arise in non-normal flight situations and operations, and they are often related to conditions that may generate inefficiencies in the airspace, result in increased workload for pilots and Air Traffic Control (ATC), or even lead to an unsafe situation. Therefore, dealing with anomalies is a natural step for achieving the goals regarding operational safety and efficiency. The modeling and discovery of anomalies shed light on hazardous or inefficient operations and substantiate the development of new safety and operational policies and practices for airlines and ATM.

There are various methodologies for anomaly detection within the flight data context. Over the past years, several studies on this matter have been performed, and they are analyzed more thoroughly in Section 2. Nevertheless, they all discuss study-specific steps for solving a particular formulation of the anomaly detection problem. Therefore, an extensive framework for the application of anomaly detection techniques in flight operations data is lacking. Furthermore, previous studies focus on methods for identifying anomalies and do not provide systematic tools for investigating and pointing out contributing factors to the anomalous situation. It is worth mentioning that explanation of machine learning models has been demonstrating significant importance to build trust in automation for improved usability in decision-making. Finally, with the ongoing development of machine learning models and techniques, there are still opportunities for applying novel or previously unexplored modeling approaches with proper comparison to baseline methods or current practice.

This paper presents a comprehensive two-module framework for anomaly detection and explanation in flight operations data, with an in-depth discussion, proposition, and demonstration of a reusable pipeline for model building and application. The anomaly detection module focuses on the development of data-driven models for the automatic identification of relevant anomalies without the need for an a priori specification of events and empirical thresholds. A subsequent anomaly explanation module focuses on the development of explanatory models that shed light on the most likely causes of flagged anomalies. We demonstrate the applicability of the proposed framework in two scenarios of routine flight operations monitoring considering both airline and air traffic management perspectives. In the first one, we explore aircraft performance data for landing operations at Minneapolis-Saint Paul International Airport (KMSP) and performed anomaly detection with an unsupervised learning approach using a density-based clustering algorithm (DBSCAN - Density-Based Spatial Clustering of Applications with Noise), while comparing the results with the current industry practice of exceedance detection. In the second application, we explore surveillance data for terminal airspace operations at Sao Paulo/Guarulhos International Airport (SBGR) with the application of isolation forests for anomaly detection in arrival trajectories, enabling the creation of an autoencoder classifier for offline anomaly detection. Additionally, supervised learning models are developed for anomaly explanation. With the application to two distinct scenarios of routine airline and ATM operations monitoring, we show how the framework is able to identify relevant anomalies (not detected by currently used methods) and provide novel insights about contributing factors to inefficiencies and potential unsafe events. For instance, with the construction of explanatory models, we learn that the anomalous situations at SBGR are more likely to be associated with landing operations on runway 27 under wind scenarios, with an increase in the odds ratio of 62% and 58% for tailwinds and headwinds, respectively.

The structure of this paper is as follows: Section 2 presents the anomaly detection problem and reviews the related literature, as well as remarks regarding the anomaly detection process and current practice. Section 3 discusses the proposed anomaly detection framework. Section 4 presents applications of the framework and discusses its impacts, and Section 5 details the conclusions.

2. Background and literature review

In this section, we first discuss general aspects of the anomaly detection problem, such as taxonomy and commonly used methods found in the literature. We then discuss the anomaly detection problem within flight operations, outlining the process, its objectives, particular challenges, approaches to problem formulation, as well as current industry practice and historical and recent developments in the literature. Finally, in light of the current practice and research efforts, we analyze research gaps and shortcomings in the field and discuss the contributions of this work.

2.1. Anomaly detection: an overview

Anomaly detection consists of finding patterns on data inconsistent with the expected behavior (Chandola et al., 2009). There are various algorithms and techniques for detecting anomalies, and a thorough literature review is presented by Chandola et al. (2009). Nevertheless, some remarks regarding usual problem formulations, taxonomy, and methods are noteworthy and discussed below.

2.1.1. The anomaly detection problem and classification

Although related, there are differences between anomaly detection problems and noise removal and accommodation ones. Anomalies are valid observations with real-life relevance. On the other hand, noise is an unwanted portion of the data that may encumber the analysis leading to potentially misleading conclusions. It is often thought of as an intrinsic source of randomness in the data. Nevertheless, it can also be the result of malfunctioning sensors, for example. In a broader sense, it can be therefore interpreted as a phenomenon that distorts the data, be it in a more low-level random measurement error or as a more high-level, "semantic" noise. In any case, there may be a need to remove, correct, or accommodate it for the studies to be conducted with valid, consistent, and robust data.

Similarly, it is also relevant to differentiate between anomalies, outliers, and novelties. Albeit related, especially regarding the discovery process, there might be nuances, according to the domain. Anomalies are observations that deviate from a sense of normality. An outlier, on the other hand, can be thought of as a rare normal instance. Finally, novelties are normal instances that had not been observed. Bringing the definitions to the flight operations domain, if we consider an example of approach procedures, an anomaly could be a high-energy approach compared to a defined normal range. An outlier could be an uncommon approach procedure of a different type of aircraft with few operations: even though it is normal, it is different from the rest. Finally, a novelty could be a newly implemented approach procedure that had yet to be captured on data. Nevertheless, despite the potential semantic differences between anomalies, outliers, and novelties, it is worth mentioning that uncommon observations that do not follow any pattern sometimes are referred to as noise. This is the case in the context of clustering analysis, for example.

Outside of the aviation domain, a few examples of anomaly detection problems are anomaly detection in astronomical data (Lochner and Bassett, 2021), medical images (Wei et al., 2018), and cyber-intrusion detection (Alqahtani et al., 2020).

Anomaly detection problems can be classified based on data input, model supervision type, and anomaly category. The input data, in turn, can be subgrouped based on the number of variables (univariable or multivariable), variable type (binary, categorical, or continuous), and observation type (independent – or point data –, or dependent, which may be a time series, sequential, or spatial data).

When building a model for detecting anomalies using statistical learning methods, the approach can be supervised, semi-supervised, or unsupervised, regarding the supervision type.

A supervised model is the one for which we have an associated response y_i for each set of predictors x_i. In the supervised anomaly detection problem, both normal and anomalous data are fed to the model as it learns to differentiate between them. For the unsupervised approach, there is no categorization of the input data as normal operations nor anomalies. Formally, there is no associated response y_i for each set of features x_i. Anomalies are detected solely based on the differences found in the data itself. Finally, semi-supervised modeling formally refers to the setting in which there are associated responses for only a portion of the observations. A semi-supervised learning model incorporates both the measurements for which there is an associated response available and those for which there is not. Within the anomaly detection context, the literature commonly refers to models built with normal data exclusively as semi-supervised learning models. However, some consider this to be confusing terminology (Ruff et al., 2021). Even though the inference is to be made concerning qualities not featured during training (anomalies), these could be seen, in practice, as supervised learning models for normal operations, as they may not incorporate data without its respective response.

Finally, anomaly detection problems can be classified based on the nature of the anomaly itself (Chandola et al., 2009). The anomalies can be classified into the following classes:

- Instantaneous, or point anomalies: individual observations are considered anomalies based on comparison with the rest of the data;
- Context-based: observations are anomalous depending on where they locate in the data, and might not be anomalies in other contexts;
- Pattern-based, or collective anomalies: a group of related observations is considered anomalous if it differs from patterns observed in data.
- Correlation-based: observations are defined as anomalous based on their correlation (Li and Hansman, 2013). It is applicable when the relationship between variables is the object of interest.

It is important to note that the classifications are not mutually exclusive. Anomalies can be pattern and context-based, or instantaneous and context-based, for example. Table 1 summarizes the categorization of anomaly detection problems.

Table 1
Anomalies categorization.

Problem aspects	Categories

Input data	Binary, categorical, continuous;
	Univariable, multivariable;
	Point data, structured data
	(time series, sequencies, spatial data).

Supervision type	Supervised, semi-supervised, unsupervised.
Anomalies	Context-based, instantaneous, pattern-based, correlation-based.

2.1.2. Anomaly detection methods

The anomaly detection problem does not have a universal solution, applicable to all cases. In practice, the existing techniques solve specific formulations of the general problem (Chandola et al., 2009).

Defining an anomaly detection technique applicable to one of these specific formulations depends on the particularities of the domain of application, the way the problem is framed, as well as the characteristics of the problem, where the aforementioned categorizations come in place, combined to the application of knowledge and tools from other areas, such as machine learning or statistics.

Over the literature, one finds several categories of anomaly detection methods, models, and tools. There is also different taxonomy for them, which can be classified based on the algorithm, principles, or the underlying tool. For example, a solution that uses autoencoders can be classified as a reconstruction-based technique or as a neural network-based one. The main techniques for anomaly detection are discussed in what follows.

Classification-based methods. Classification-based techniques work on the assumption that it is possible to build a classifier (or model) that can distinguish between anomalous and normal cases based on the sample space. A model is constructed using a series of categorized data (training phase) so that, during testing and operations phases, it is able to classify a new instance between the anomaly or normal categories (Chandola et al., 2009).

Nearest neighbor methods. The fundamental hypothesis for the nearest neighbor techniques is that normal data locates itself in dense neighborhoods, while anomalies are distant from their closest neighbors. The choice of how to measure distance varies — from Euclidean distance to probability-based distance measurements. A disadvantage is that, if there are normal samples in data that do not have enough close neighbors, or if there are anomalies with close neighbors, then the technique may fail to point them out correctly.

Clustering methods. Clustering techniques involve grouping similar samples of data into a cluster. According to Chandola et al. (2009), there are three categories of clustering-based anomaly detection techniques. For the first technique, the idea is that normal data belong to clusters, while anomalies do not belong to any cluster. An example of an algorithm belonging to this first category, which does not require for every instance to belong to a cluster, is DBSCAN (Ester et al., 1996). The second category of clustering-based techniques is supported by the idea that normal data are closer to the centroids of the nearest clusters, while anomalies are far from the centroids of their respective nearest clusters. In this case, if all anomalies form a cluster among themselves, the technique would not be able to identify them. Lastly, the third approach is based on the assumption that normal data belong to large and dense clusters, while anomalies locate themselves in small or dispersed clusters.

Statistical methods. Statistical techniques for anomaly detection are based on the assumption that normal data occur in high probability regions of a stochastic model, while anomalies occur in low probability regions (Chandola et al., 2009). Based on a stochastic model constructed from data, this approach involves performing an inference test to determine whether an observation belongs to the model or not: data with low probabilities of having been generated by the model are considered anomalies. One of the advantages of the statistical approach is that the detection of anomalies is associated with a confidence interval, which can help substantiate a decision. Another advantage is that this approach can be unsupervised, given that there is a robust process for estimating the distribution (Chandola et al., 2009). The main disadvantage here is that this approach relies on being true the hypothesis that the data is generated by a specific distribution. According to Chandola et al. (2009), this is often not the case.

Spectral methods. Spectral methods for anomaly detection rely on the assumption that, once the data is transformed into a lower-dimensional subspace, then anomalies and normal observations manifest themselves differently. They are also referred to as subspace-based methods. One of the most used tools in this category is Principal Component Analysis (PCA). Nevertheless, spectral methods can be coupled with any other tools, when the anomaly detection task is then performed in the desired subspace.

Reconstruction methods. Reconstruction methods rely on models that learn how to reconstruct normal data. When the model fails to reconstruct an observation – defined by comparing the reconstruction error to a previously tuned threshold –, that observation is considered an anomaly. PCA is often applied in this approach, as in the spectral methods. The principal components, however, are reconstructed to the original sample space. Autoencoders are also commonly used within the reconstruction methods.

Isolation methods. Isolation methods aim at separating an instance from the rest. It measures the susceptibility of each instance to be isolated, with the anomalies being those more easily isolated. It relies on the principle that anomalies are few and different than normal instances. Two examples of this approach are the Isolation Forest – discussed in Liu et al. (2008) and Liu et al. (2012) – and the Extended Isolation Forest (Hariri et al., 2021). Both approaches rely on a binary tree structure called iTree to isolate the instances.

2.2. Anomaly detection in flight operations

When considering the anomaly detection problem in the flight operations domain, one deals with data consequential of activities and interactions between two major participants: aircraft operators and ATM.

Flight operations data include primarily aircraft performance data from on-board sensors (flight data recorders) and flight tracking data from surveillance equipment (radar, ADS-B). This core information can be complemented with other sources, e.g., safety reports, flight plans, weather data, workload information, and airspace structure, enabling the analysis to be as thorough as one desires.

Below we discuss the anomaly detection process in flight operations and detail its goals before also dealing with challenges particular to this domain and research efforts.

2.2.1. Anomaly detection process and current practice

Based on the available information, the anomaly detection process emerges. It can be thought of as two parallel, complementary, and self-interacting approaches: offline anomaly detection, and online anomaly detection. Furthermore, the approach in question dictates the way one thinks about and deals with the flight operations data and anomaly models. The models can be expected to work with complete data, so it can be applied in the offline process, but it can also be compatible with streaming, incomplete data, which enables online anomaly detection. Fig. 1 illustrates the anomaly detection process and its two principal fronts.

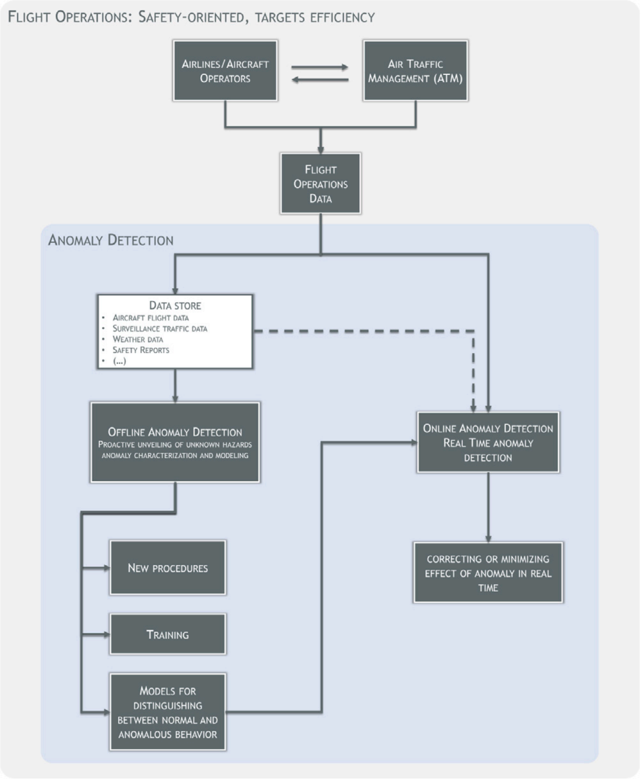
Flight Operations: Safety-oriented, targets efficiency
Airlines/Aircraft Operators
Air Traffic Management (ATM)
Flight Operations Data
Anomaly Detection
Data store
• Aircraft flight data
• Surveillance traffic data
• Weather data
• Safety Reports
• (...)
Offline Anomaly Detection
Proactive unveiling of unknown hazards
anomaly characterization and modeling
Online Anomaly Detection
Real Time anomaly detection
New procedures
Training
Models for distinguishing between normal and anomalous behavior
Correcting or minimizing effect of anomaly in real time

Fig. 1. Anomaly detection process.

Offline anomaly detection. When talking about offline anomaly detection, the live, streaming flight tracking data and the downloaded aircraft performance FDR data are saved in data repositories for usage after the operation has happened, in a process commonly referred to as batch data processing. Offline anomaly detection is particularly useful for the proactive unveiling of unknown hazards, and anomaly characterization and modeling.

From the offline approach, there may stem the development of novel safety policies with new operational procedures; training programs to prevent the anomaly from happening again; models for distinguishing between normal and anomalous behavior; and the discovery of precursors — i.e., correlated events that occur before the anomaly.

Online anomaly detection. The second approach is online anomaly detection. Here, one deals with the real-time anomaly detection task with the primary goal of correcting or minimizing the effect of the anomaly during the operations.

In this scenario, the flight data is streamed into the processing engine, and the models deal with the unfinished, developing data. This is hence referred to as streaming data processing. Nevertheless, the online model can also benefit from the offline data from the previous operations, stored in the data repositories.

Furthermore, online anomaly detection may leverage the findings of the offline anomaly detection process, regarding the characterization of the anomalies themselves or the mining of precursors.

Current practice. From the airline standpoint, current practice comprises the scenario-oriented approaches (Höhndorf, 2018) of Flight Operations Quality Assurance (FOQA) programs, also known as Flight Data Monitoring (FDM), and regards the comparison of sensor data with pre-established thresholds. For that reason, it is also known as exceedance detection. It is this approach that the Federal Aviation Administration (FAA), for example, details in the Advisory Circular

(AC) 120-82 - Flight Operational Quality Assurance, aimed at presenting means of developing and implementing an acceptable FOQA program for the authority (FAA, 2004). Furthermore, in terms of analysis horizon, current practice occurs solely offline.

On the other hand, there is not a well established methodology and common practice for anomaly detection and monitoring of flight trajectories in the ATM domain.

2.2.2. Challenges

The challenges regarding anomaly detection problems in the aviation domain are manifold. First, flight operations data – from aircraft recording devices and ATM surveillance equipment – is inherently high-dimensional. Even though one could formulate an anomaly detection problem based on, e.g., safety reports, the analyses stem largely from sensor data recorded in a time series. To illustrate the high-dimensionality of the data, if we consider a DFDR compliant with the requirements of 14 CFR Part 135.152 Flight data recorders (Federal Aviation Regulations, 2009), at least 88 parameters are being recorded. If they are sampled at a rate of up to 8 Hz, that translates into 28 801 records per parameter after a one-hour flight. Moreover, modern recording devices register over a thousand parameters – that may be at even higher sampling rates –, the average commercial flight duration is over one hour, and the analyses evaluate thousands of flights. There is then a large pool of high-dimensional data from which to choose. Even if one formulates the problem based on a subset of the data or even summarizing metrics, it still poses a challenge regarding feature selection and engineering.

A further challenge is the noise typically present in flight data. This creates the need for being able to distinguish between anomalous behavior and observations that do not follow any pattern (unstructured data portion). Nonetheless, even with noise accommodation procedures, it is possible to find unwanted records with errors within working data.

Another issue is data set contamination or pollution with equivocally categorized data. It is often the case that some observations labeled as and initially believed to be normal are, in fact, anomalous, making the distinction task more challenging. In aviation, this is aggravated by the high number of flights and the lack of established anomaly detection programs, along with the current practice of identifying only a specific set of previously defined events.

The multi-scale aspect of aviation data can also be challenging. The data portrays different contexts of the flight – with distinct flight phases – sampled at different time rates.

Finally, there is also the managerial aspect of the problem. The anomaly detection task in aviation highly relies on domain-expert analysis and subsequent capability to investigate the potential scenarios. Hence, the anomaly detection process must highlight a manageable number of potential anomalies compatible with the available person-hours for any further investigation.

2.2.3. Approaches to the formulation of an anomaly detection problem in flight operations

There are multiple ways to frame a single, broad anomaly detection problem. Within the flight operations context, it could be thought of as a comparison of different summarizing metrics, as a comparison of unique sequential data (e.g., time series), or as a single time-series anomaly detection problem, among others. Therefore, even though the starting point is a collection of time-series sensor data, one can formulate the problem in different formats. Multiple techniques and algorithms can then be used and have been so, as discussed in the following sections regarding the applications found in the literature.

Figs. 2 and 3 and Table 2 exemplifies these different ways of framing the anomaly detection problem.

Table 2
Anomaly detection problem sample formulation: summary metrics.

Flight	Vertical acceleration	Glideslope deviation	...	Pitch
$flight_1$	n_{z_1}	δ_{gls_1}	...	θ_1
$flight_i$	n_{z_i}	δ_{gls_i}	...	θ_i
$flight_n$	n_{z_n}	δ_{gls_n}	...	θ_n

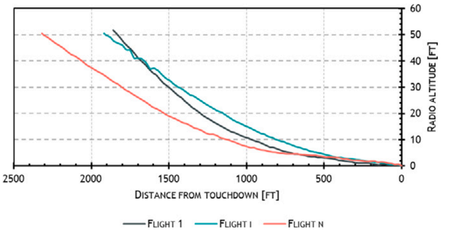

Fig. 2. Anomaly detection problem sample formulation: comparison of radio altitude for multiple landings as independent distance series.

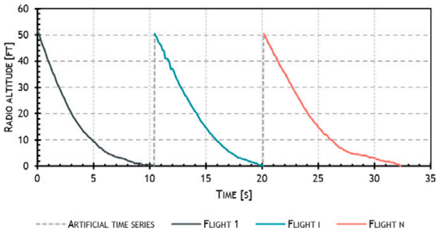

Fig. 3. Anomaly detection problem sample formulation: comparison of radio altitude for multiple landings as a single time series problem.

2.2.4. Anomaly detection from the airline standpoint: previous research efforts

One of the first initiatives to detect anomalies on flight data is that of Amidan and Ferryman (2000). In this study, the authors applied Principal Component Analysis (PCA) via Singular Value Decomposition (SVD) to search for the atypical flights. The research introduced key concepts and methodologies regarding this branch of anomaly detection. According to Amidan and Ferryman (2000), the method succeeded in ordering flights according to their degree of atypicality quickly and efficiently.

Another precursor work was that of Amidan and Ferryman (2005). In this one, a software called "Morning Report" was developed to find typical patterns and atypical events in data. This paper introduced the idea of applying clustering algorithms in the context of anomaly detection in aviation. The approach taken would allow the specialist to focus on atypical flights, a concept used until today. Nevertheless, this approach proved to be limited: the time series representation, for example, failed to capture relevant signals in the data (Li and Hansman, 2013).

Das et al. (2010) proposed an anomaly detection approach based on Multiple Kernel Learning (MKAD), focused on detecting anomalies in multivariable, high-dimensional data. According to Das et al. (2010), the developed algorithm was able to detect anomalies that were not captured by the state of the art at the time.

Gorinevsky et al. (2012) developed a method for detecting anomalies based on a regression model, aiming at assisting FOQA processes. The model was tested on flight data and was able to identify anomalous records. However, the anomalies found were more related to faulty sensor readings than safety anomalies per se.

Li and Hansman (2013) developed an anomaly detection approach based on clustering algorithms and expert review. The authors developed two algorithms: ClusterAD-Flight and ClusterAD-Data Sample. As the expert review was one of the research pillars, it also focused on developing visualization tools. ClusterAD-Flight used DBSCAN (Ester et al., 1996) as the clustering algorithm due to its capabilities of automatically determining the number of clusters, treating noise in data, and detecting extreme values as it identifies the clusters. ClusterAD-Data Sample, on the other hand, identifies flight data sample clusters using Gaussian mixture models. For analysis, ClusterAD-Flight converts each flight into a vector in hyperspace, while ClusterAD-Data Sample converts each sample obtained from the recorder into a vector. As per (Li and Hansman, 2013), the two algorithms were capable of identifying operationally relevant anomalies, surpassing current methods.

Another recent effort regarding anomaly detection applied to aviation is that of Fernandez et al. (2019). They focused on detecting anomalies during the approach phase, applied to a 35 000-approach data set on runway 25R in LEBL airport. According to the authors, their methodology consisted of two phases. First, during what they named descriptive analysis, they performed clustering analysis using the HBDSCAN algorithm for classifying the flights as normal and abnormal. Second, they used the normal flights to train a neural network – more specifically, an autoencoder – that reconstructs a given approach. This reconstructed approach is then compared to the actual approach (input), which outputs an error metric. Their fundamental hypothesis is that, since they trained the autoencoder for reconstructing normal approaches only, these would return a low reconstruction error, while the anomalous flights would output a high error. After defining an empirically set threshold for the reconstruction error, the autoencoder can classify new approaches as normal or abnormal. According to Fernandez et al. (2019), the autoencoder was able to classify more than 74% of the anomalies, struggling most with the near-normal ones. Finally, they envisioned an implementation of it as an automatic labeling system that aids safety analysts and claimed the feasibility of automatically filtering outliers.

2.2.5. *Anomaly detection from the ATM standpoint: previous research efforts*

Matthews et al. (2013) presented an approach for discovering operationally significant anomalies in data generated from surveillance equipment. It is an extension of the studies in the FOQA/FDM domain to ATM data. It used the MKAD algorithm presented by Das et al. (2010) to identify anomalies regarding the flight track. The algorithm identified approximately 40 anomalous flights, which were then further analyzed. Domain experts confirmed operationally significant anomalies in 15 of these flights.

Murça (2018) discusses anomaly detection via Conformal Prediction for identifying non-conforming trajectories within the proposed framework for characterization of air traffic flows based on flight trajectory data. The Conformal Prediction model presented better values for recall, precision, and F1-score when compared to K-Nearest Neighbors (KNN) and Gaussian Mixture Model (GMM).

Deshmukh and Hwang (2019) presented TempAD, an unsupervised learning algorithm that uses temporal logic for anomaly detection for terminal airspace operations. It generates normal-flight parameter intervals that are easily interpreted and converted to natural language.

Subsequently, Deshmukh et al. (2019) presented an approach to identify precursors for detected anomalies in surveillance data for terminal airspace operations. For that, the authors presented a supervised learning algorithm for precursor detection, Reactive TempAD. Deshmukh (2020) later extended the results obtained for data-driven anomaly and precursor detection in metroplex airspace operations.

Another effort is that of Olive and Basora (2019), which presented a methodology to analyze flight track data from ADS-B and identify operationally significant anomalies. The authors obtained the principal flows in the airspace via trajectory clustering and used autoencoders for identifying anomalies.

2.2.6. *Research gaps and shortcomings of current practice*

Current literature discusses study-specific steps for solving particular formulations of the anomaly detection problem. Therefore, an overarching framework for the application of anomaly detection techniques in flight operations from which one can generate specific pipelines is lacking. Furthermore, recent research focuses on methods for identifying anomalies and does not provide systematic tools for investigating contributing factors to the anomalous situation. Explanation of machine learning models has been demonstrating significant importance to build trust in automation for improved usability in decision-making, but has been typically overlooked in previous work related to anomaly detection in the flight operations domain. Finally, with the ongoing development of machine learning models and techniques, there are still opportunities for applying novel or previously unexplored modeling approaches.

In terms of current practice in the airline subdomain, since it relies on the specification of thresholds of previously defined safety events, the latent risks and the near-exceedance situations that did not lead to an event go unnoticed. Additionally, risks that were not even mapped also go undetected — only what is sought, known, and searched for is captured and ends up being described. Furthermore, because the current approach is safety exclusive, current practice does not benefit from anomaly detection methodologies to assess operational efficiency and aid in the discovery of improvement opportunities. For ATM, the aforementioned lack of common procedures with a well-established methodology for anomaly detection and monitoring of flight trajectories hinders the systematic evaluation of the system performance.

This work addresses these gaps with the proposition of a systematic framework for anomaly detection and explanation in flight operations data that does not depend on the previous specification of heuristics. The anomaly detection module of the framework focuses on the development of data-driven models for the automatic identification of relevant anomalies, without relying on events and thresholds empirically defined. A subsequent anomaly explanation module focuses on the development of explanatory models that shed light on the most likely causes of flagged anomalies, towards improved trust and usability of the machine-based anomaly detection tool for decision-making. With the application of the framework to two scenarios associated with routine airline and ATM operations monitoring using distinct data types, we demonstrate how the proposed framework can serve multiple use cases in flight operations monitoring and contribute to the enhancement of the current practice. In the first application, the framework was able to automatically identify relevant aircraft performance anomalies during landing operations, which were not detected by currently used exceedance-based methods. In the second application, the framework allowed for the automatic discovery of flight trajectory anomalies in terminal airspace, while identifying their most likely causal factors, providing novel insights regarding the relation between operational factors and inefficiencies at the airport of study.

3. Methodological approach

To fulfill our goals of laying out an overarching approach for anomaly detection in flight operations as a whole as well as providing systematic tools for investigating anomaly contributing factors, we divided our work into two fronts: definition of a framework with step by step remarks on data processing and model building; and the application of the proposed framework within two operational scenarios with the comparison of the anomaly detection approach with baseline methods, investigation of anomaly contributing factors, and the novel application of isolation forests in the flight operations context.

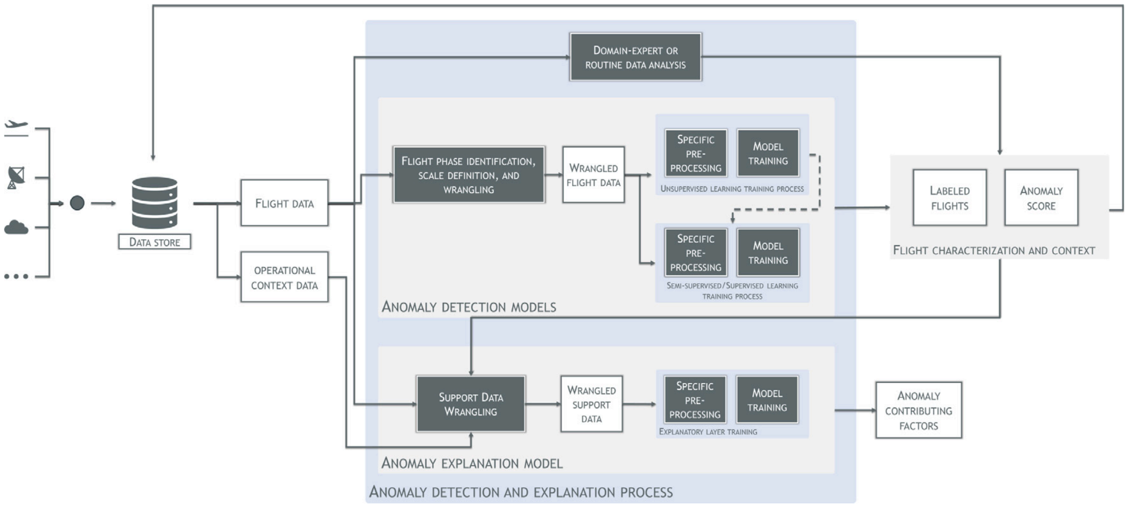
Domain-expert or routine data analysis
Data store
Flight data
Operational context data
Flight phase identification, scale definition, and wrangling
Wrangled flight data
Specific pre-processing
Model training
Unsupervised learning training process
Specific pre-processing
Model training
Semi-supervised/Supervised learning training process
Anomaly detection models
Labeled flights
Anomaly score
Flight characterization and context
Support Data Wrangling
Wrangled support data
Specific pre-processing
Model training
Explanatory layer training
Anomaly contributing factors
Anomaly explanation model
Anomaly detection and explanation process

Fig. 4. An overview of the proposed anomaly detection framework: workflow and architecture.

3.1. Anomaly detection framework

For the definition of the framework, we consider first a black-box view of the models to develop an architecture that is comprehensive yet general enough that allows for the application of different kinds of anomaly detection models in both offline and online contexts while also leveraging domain expert analysis and knowledge acquired in currently established processes such as FOQA. We treat the framework concept in a similar fashion to the one in software engineering. It is not a fixed procedure but rather an abstraction of the domain-specific process, adaptable in accordance to particular anomaly detection problems in the flight operations area. Fig. 4 displays the proposed framework, discussed in further detail below.

We divide the framework into two fronts: anomaly detection, and anomaly explanation. During the anomaly detection model conception, one formulates the problem via reproducible data wrangling and preprocessing steps and then proceeds to train unsupervised, supervised, or semi-supervised statistical learning models. Then, based on the labeled flights and anomaly scores, one uses the learned characterization within an explanatory pipeline that unveils possible causes for a given anomaly.

3.1.1. Data streaming and storage

The starting point of the framework is to gather and organize the data to be used throughout the process. As the data is generated during flight operations, it is streamed or loaded into the data analytics pipeline. Regarding its storage for posterior usage, data might be kept in a data store. The major source of information is flight data — surveillance or aircraft recorder data. In addition, other sources of information can also be used, e.g., weather data, safety reports, and flight plans. In this sense, for modeling purposes, we can also categorize the data between flight data per se, and the operational context data to be used during the explanatory phase.

3.1.2. Leveraging domain-expert routine analysis and established practice

The first way to identify anomalies in the data is via domain-expert, routine data analysis. In airlines, it is very common and often mandatory to have a FOQA/FDM program instituted. It is current practice for the FOQA analyst to analyze flight data, whether it is from incident investigation or analyzing operational procedures. During this routine data analysis, an anomalous situation could be identified in the data. From this method, it is thus possible to obtain labeled anomalous flights or operations and the associated context. This labeled data can then be used to build semi-supervised and supervised learning models, discussed in more detail in Section 3.1.3.

3.1.3. Anomaly detection models

The second way to identify anomalies in flight data is via the data mining approach, which starts with model conception. During this phase, one formulates the problem via reproducible data wrangling and preprocessing steps and then proceeds to train unsupervised, supervised, or semi-supervised statistical learning models.

Flight phase identification, scale definition, and data wrangling and processing. The first step in the data processing sequence when building the anomaly detection model is the flight phase identification. It allows for the reduction of problem dimensions and the definition of the analysis scale. Flight phase identification is usually performed given a set of heuristics based on sensor data - e.g. pressure and radio altitude values, aircraft heading, airspeed, and rate of climb.

During data wrangling and processing, the necessary problem-specific operations are performed. One could join the different data sources, filter the data to specific periods of the flight, impute missing values, select specific features, calculate derived variables, identify the flight phases, calculate summarizing metrics, and more. The output of this step is either having the data formatted in accordance with the problem conception or having it in a way that enables this formatting during the specific preprocessing phase. In addition, the steps performed for model conception must be repeatable at the time of model operation.

As mentioned on Section 2.2.3, there are multiple ways to frame the anomaly detection problem. In that respect, the data wrangling step is one of the most important ones of the pipeline as it prepares the data for solving the posed problem.

Unsupervised learning approach. For the unsupervised learning approach, based on the unlabeled processed flight data, the identification of anomalies is performed by exploring the structure of the data itself. By analyzing the samples with respect to themselves, the goal is to find the ones that do not follow the normal patterns of the data.

The unsupervised learning training process is composed of two steps. First, there is a specific preprocessing phase. If we are dealing with distance-based clustering models, for example, this is where we would scale the data and select/resample it to standardize the dimensions.

The second step is the model training per se. Given the chosen model, this is where one finds the adequate hyperparameters that yield a model with the desired performance metrics.

Semi-supervised/supervised approaches. The second statistical learning approach comprise the semi-supervised or supervised learning training process. Given the labeled data obtained via the domain-expert routine analysis and the unsupervised learning process, one can now develop a statistical model that identifies anomalies leveraging the labeled normal flights, or labeled anomalous and normal flights. If the model uses only labeled data during the training process, it is a supervised learning model, while a model that uses both labeled and unlabeled data is usually referred to as semi-supervised.

Similar to the unsupervised learning approach, there is first a specific preprocessing phase. Once again, it is followed by the model training per se: given the chosen model, one finds the adequate hyperparameters that yield a model with the desired performance metrics.

3.1.4. Anomaly explanation model

The anomaly explanation comprises training a supervised learning model that either classifies an observation as anomalous or normal or predicts its anomaly score, given the support data. The pipeline starts with the wrangling of the supporting data, containing information on the previously assessed anomalous qualities of a given flight; a selected subset of the operational context data – e.g., weather conditions – ; and may also include the flight data itself. From the model, one can obtain explanatory metrics such as SHAP (SHapley Additive exPlanations) (Lundberg and Lee, 2017), feature importance values, or linear or logistic regression coefficients, thus shedding light on anomaly contributing factors.

3.1.5. Remarks on model training, selection and operationalization strategies

Although machine learning model training comprises several model-specific steps, there are noteworthy general guidelines and principles.

Model operationalization. Once the models are trained, they can be extracted and served in a model operation process. It should be noted that the data format established during the data wrangling and processing phase for model conception must be re-ensured at the time of operation.

As the served models flag new flights as anomalies during operations, these can then go through an investigatory and validation process by a domain expert. This process adds labeled flights and contexts to the collection of labeled anomalies and normal instances. This data can be kept in a data store, fed back to improve the semi-supervised or supervised learning models, and even used at the time of operation as a comparison reference to the newly flagged anomalies, for example. In this sense, the constant investigation and evaluation of flagged anomalies lead to the reassessment of model performance. To account for the updated information, models will be retrained and served in their operational pipelines multiple times. In this way, solid procedures for continuously training, monitoring, and serving the models are essential.

These concepts fall under the field of MLOps and are decisive for the successful implementation of continuous anomaly detection initiatives via machine learning.

Train, validation and test splits. It is often adequate to split the available data set into train, validation, and test subsets. Doing so enables model selection and performance assessment on data not used during training. Although there is no single nor absolute way to do it, train–validation–test ratios of 75:15:15 or 60:20:20 are common. Because anomalies are few and might be diverse – which leads to a class imbalance scenario – special attention is needed in this step, if performed.

Feature scaling. Commonly, one needs to scale the data set features before applying a machine learning model. The scaling process may contribute to the proper assessment of feature importance by the model. There are multiple scaling techniques, such as standardization, normalization, robust scaling, and absolute value-based methods. It is important nonetheless to ensure the scaling principle matches the desired application. Scaling methods robust to outliers, for example, if applied to anomalies, may filter out observations that would be flagged as anomalous by the model, making them go unnoticed. The same technique however can be beneficial when training a model representative of normal operations. As it naturally deals with outliers, robust scaling may lessen data set contamination effects. In this manner, when transforming anomalous data based on a robust scaler trained with normal data, the anomalous qualities of the data may become clearer.

Performance metrics and model selection. Another common challenge is assessing whether a model is adequate and how to choose among the trained candidates. Whereas typical performance metrics such as precision, recall, F1-score, AUROC (Area Under the Receiver Operating Characteristics), and others help, they must be analyzed within the model usage context.

Furthermore, one must evaluate the performance metric under the proper operational constraints. If a team is capable of analyzing k potential anomalies in a given period, the metric must be considered under this limitation. In general terms, an adequate strategy is to evaluate a given $metric@k$, where k is the response capacity constraint.

There is not nonetheless a general rule for selecting a suitable performance metric. An appropriate choice relies on the entire operational context. In an online model, for example, for which there is an expected response in case of a flagged anomaly, it might be the case for the model to prioritize precision in a way that it minimizes false alarms (type I error) that could lead to unnecessary, or wrong interventions. If the responsible team is capable of analyzing ten potential anomalies in a given period, we would want to maximize precision within the ten flagged anomalies, making the metric $precision@10$, in this scenario.

If we consider offline anomaly detection, it might be the case to prioritize a more balanced system. In this scenario, the evaluation of the *F1-score*$@k$ can be adequate. The metric could also be directly related to a business goal. For an airline with an established FOQA/FDM program, for example, it could also be appropriate to evaluate the number of anomalies that otherwise would be unnoticed - $otherwise\_unnoticed\_anomalies$. The metric would therefore be $otherwise\_unnoticed\_anomalies@k$. However, when evaluating proactive anomaly discovery, the initiative alone is often more important than a performance metric. The detection of one operationally relevant anomaly that would otherwise go unnoticed may be satisfactory by itself, especially considering the transition to a more proactive safety management model, commented on in the Introduction.

Finally, it is good practice to elaborate and ponder on the most appropriate metrics before moving on to model conception, as well as evaluate the proper trade-off between false alarms and missed anomalies.

Model ensemble and model breakdown. As mentioned in Section 2.2.2, one of the challenges when considering the anomaly detection problem in the aviation domain is the multiple scales in which one can represent the same data. Moreover, anomalies manifest themselves in several ways and lack uniform representations. In this sense, one strategy to deal with these challenges is to train several smaller, more specialized models for anomaly detection. Instead of building a single monolithic model, it is possible to break the problem down between its various scales and data and operational contexts. While Agarwala et al. (2021) suggests that a monolithic model can learn across multiple, diverse domains, they also mention the increased interpretability and reduced computation cost that might come with modular architectures. Finally, even within a sole context, a valid strategy is to develop multiple models, with different techniques, and arrange them in a model ensemble.

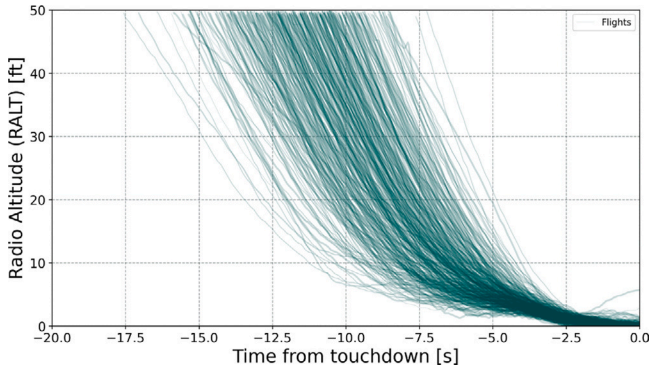

Fig. 5. Radio altitude versus time until touchdown for landing operations at KMSP.

3.2. Framework application process

To demonstrate the applicability of the proposed framework, we evaluate two settings in the flight operations domain. In the first one, we analyze landing operations at the Minneapolis-Saint Paul International Airport (KMSP) within an unsupervised anomaly detection pipeline with DBSCAN (Density-Based Spatial Clustering of Applications with Noise) and compare the results to the current industry practice of exceedance detection. In the second formulation, we analyze surveillance data for terminal airspace operations at Sao Paulo/Guarulhos International Airport (SBGR) with the application of an intermediate isolation forest model that selects normal data and feeds the training process of an autoencoder classifier for offline anomaly detection.

As mentioned before, we treat the proposed framework as an abstraction of the domain-specific broad pipeline, adaptable in accordance to problem specificities, a similar idea to the framework concept in software engineering. In this sense, for each application, we construct a particular pipeline comprising the actual data processing and model training steps. Section 4 presents discussions about the raw data for each scenario, the detailed pipelines, and results regarding the identified anomalies, as well as the comparison with benchmarks and the association between anomalies and the operational context.

4. Results and discussion

4.1. Anomaly detection in airline landing operations at KMSP via aircraft performance data analytics

For the airline operations application, we developed a model conception case study evaluating 404 landing operations at the Minneapolis–Saint Paul International Airport (KMSP). The case study is a simplified version of the problem and considers only the radio altitude parameter (RALT) within an unsupervised learning approach with DBSCAN for a offline anomaly detection formulation. Fig. 5 presents the lateral profile of the evaluated landing operations, in terms of the radio altitude parameter, while Fig. 6 presents the conceptual schematic of the framework application.

4.1.1. Data

The Sample Flight Data data set was used, available in NASA's Discovery in Aeronautics Systems Health (DASHlink) (2012). It is a publicly available data set containing flight data of a single regional jet model recorded during commercial operations over three years. The complete data set comprise more than 180,000 files, each containing records of a single flight or ground operation, grouped in a series of compressed master files.

Each flight data file presents flight information in a time series, featuring more than 100 parameters collected by several aircraft sensors. In this simplified case study, we use only the radio altitude parameter (RALT) for anomaly detection and the air–ground sensor parameter (WOW) for aiding in data subsetting in terms of flight phase identification.

We evaluated the landing operations in the following compressed master files: Tail_652_1, Tail_652_2, Tail_652_3, and Tail_652_4. Within this subset of more than 2000 flights, 404 flights landed in the Minneapolis–Saint Paul International Airport (KMSP). These are the flights that we analyze in this case study.

4.1.2. Anomaly detection model

Data wrangling and processing. Model conception starts with the specification of a data wrangling phase. In this case study, we analyzed landing operations at KMSP airport, starting when the aircraft crosses the 50 ft threshold and ending when the air–ground sensor recorded the first "GROUND". One characteristic of aircraft sensor data is that, because data come from sensors with different sampling rates and different recording times, a great part of the data is composed of missing values, if arranged in a single-index time series.

To ensure data consistency with the desired scale and deal with the missing data, the data wrangling process consisted of two steps:

1. Imputation of missing data, consisting of interpolation for the continuous parameters, and forward-filling for the discrete ones;
2. Selection of the relevant portion of the flights: from the moment the RALT parameter recorded a value less than or equal to 50 ft until the air–ground sensor first recorded "GROUND".

Unsupervised learning: DBSCAN. For building an unsupervised learning model, we used a density-based clustering algorithm named DBSCAN (Density-Based Spatial Clustering of Applications with Noise).

As suggested in the framework, the model training process consisted of a specific preprocessing phase followed by the model training per se via hyperparameter definition.

Specific preprocessing After wrangling, the data went through the specific preprocessing phase. This phase was responsible for getting the data ready for model application and turning it compliant with DBSCAN requirements and best practices.

There were four preprocessing steps:

1. Resampling of the flights so that every flight has the same dimension. Since we used the euclidean distance metric for calculating the pairwise distances for feeding the DBSCAN clustering process, every flight must be in the same hyperspace. Because every landing is different – somewhat shorter or longer – , which results in a different number of samples, it is necessary to resample – by either upsampling or downsampling – the flights to a common value. Here, we upsampled every flight to the maximum observed flight-wise number of samples;
2. Feature selection and the generation of a high-dimensional vector x for each flight. Here, for each flight, we selected the radio altitude resampled time series and arranged it sequentially in a vector. Every ith vector x_i was then grouped into a flight matrix X, used for model training;
3. Feature standardization by removing the mean and scaling to unit variance. Once again, because we are calculating euclidean distances between points, every feature must have similar scales — avoiding a distance calculation skew;
4. Dimension reduction via Principal Component Analysis (PCA), keeping 50 components of the data.

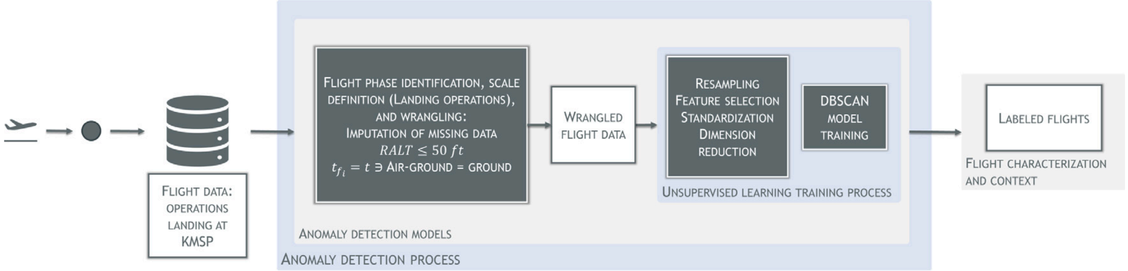
Flight data: operations landing at KMSP
Flight phase identification, scale definition (Landing operations), and wrangling:
Imputation of missing data
$RALT \leq 50\ ft$
$t_{f_i} = t \ni$ Air-ground = ground
Wrangled flight data
Resampling
Feature selection
Standardization
Dimension reduction
DBSCAN model training
Unsupervised learning training process
Anomaly detection models
Anomaly detection process
Labeled flights
Flight characterization and context

Fig. 6. Applied anomaly detection framework: landing operations at KMSP.

Table 3
FOQA event parameters and definitions (FAA, 2004).

Event name	Event description	Parameters	Event definition
Landing in a crab	An event to detect failure to align aircraft with the runway at touchdown.	Heading; Calibrated Airspeed (CAS).	Δ Heading at Touchdown vs. Average Heading until CAS = 60 knots.
Hard landing	An event that measures excessive G-force at touchdown, indicating a hard landing.	Air/Ground Switch; Vertical Acceleration.	Air/Ground = Ground, Vertical Acceleration > x G
Bounced landing	An event that measures excessive G-force at touchdown followed by a second excessive G-force, indicating a bounced, hard landing.	Air/Ground Switch; Vertical Acceleration.	Air/Ground = Ground, Vertical Acceleration > x G, followed by second Vertical Acceleration > x G within 20 s of first touchdown.

Model training DBSCAN model training relies on two major hyperparameters. The first one, ϵ, defines the distance threshold for considering two points as members of the same neighborhood. The second one, MinPts, refers to the minimum number of observations within the ϵ-neighborhood for considering a point as a core point during the clustering process.

There are multiple strategies for defining a proper value for the parameter ϵ. One of them is to cluster the data with several candidate ϵ values and assess the cluster qualities according to a given metric such as silhouettes (Rousseeuw, 1987). In this paper, we calculated ϵ in accordance with (Rahmah and Sitanggang, 2016). Furthermore, we defined MinPts as 10 to avoid the generation of small clusters. In this way, we interpreted every point labeled as “noise” as potentially anomalous. Finally, we used the DBSCAN implementation available in the open-source machine library scikit-learn, written in Python (Pedregosa et al., 2011).

4.1.3. Benchmark

For assessing the anomaly detection pipeline performance, we compare its results to the current practice of exceedance detection. For that, we evaluate three events commonly monitored for the landing phase of the flight: hard landing, landing in a crab and bounced landing. We used the recommended parameters and event definitions of AC 120-82, reproduced in Table 3. For the hard landing and bounced landing, we consider a vertical acceleration threshold of 2 g, in terms of the load factor, and for the landing-in-a-crab event, we adopt a threshold of 15 degrees in heading deviation.

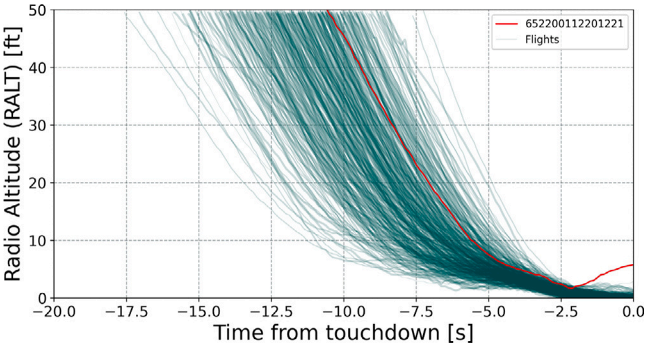

Fig. 7. Radio altitude versus time until touchdown for landing operations at KMSP with potentially anomalous flight “652200112201221” highlighted in red. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.1.4. Results

The implemented pipeline identified 20 potential anomalies, roughly 5% of the data. This number was adequate as the pipeline identified a manageable number of potentially anomalous flights without causing work overload. In addition, we were able to execute the DBSCAN pipeline without fine-tuning the hyperparameter epsilon, due to the adequate first estimate of its value.

One of the landing operations the pipeline categorized as anomalous was that of flight “652200112201221”. Fig. 7 displays the landing operations with the potential anomaly highlighted in red.

Fig. 8 displays the radio-altitude and air–ground measurements for the potentially anomalous landing operation. As seen by the air–ground parameter, this is a case of a bounced landing, which confirms the anomalous behavior of this operation. Albeit easily detected by a domain expert, it is noteworthy that manual inspection of flights, let alone the analysis down to the parameter level, is impractical in real operations. Therefore, having access to tools that flag and highlight potentially anomalous flights for further investigation is of particular importance.

Comparison with current practice. While the current practice of routine data analysis and the machine learning approach are complementary, rather than competing — with the current practice being able to provide labeled data in our framework, for instance, we compare its results to those of the proposed anomaly detection pipeline as a way to assess whether the proposed approach extends current capabilities.

The exceedance-based FOQA pipeline did not identify any of the evaluated safety events within the flights. In this sense, the $otherwise\_unnoticed\_anomalies$ metric in this scenario, in terms of ratio, is 1.

The FOQA pipeline did not identify even the bounced landing event, influenced by the defined threshold aimed at identifying events with higher loads. On the other hand, the proposed anomaly detection model identified this bounced landing event without specifying a heuristic rule

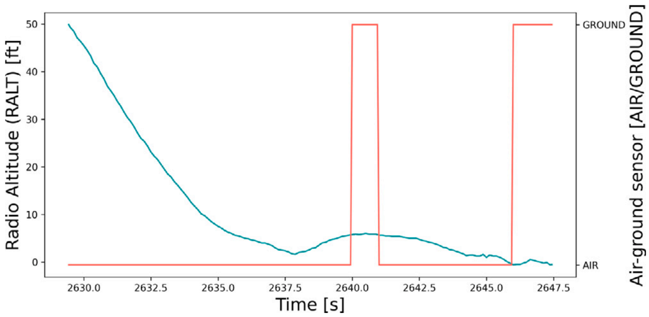

Fig. 8. Landing operation for potentially anomalous flight "652200112201221" - bounced landing.

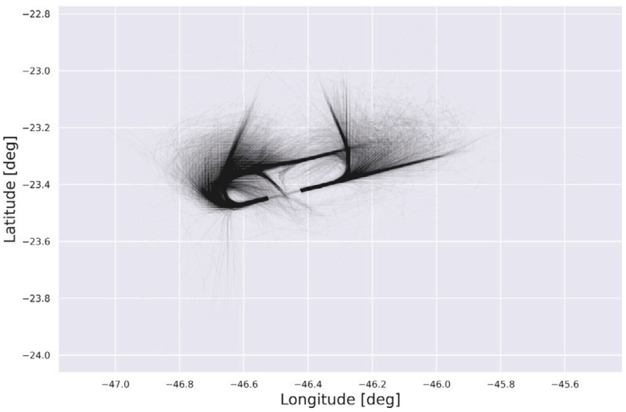

Fig. 9. Horizontal trajectories of terminal airspace arrival operations at SBGR between December 31, 2019 and January 31, 2020.

for it nor previously setting an empirical threshold. In normal operational settings, this anomaly would indeed go unnoticed since it did not reach the threshold for the required severity level. Nevertheless, the automatic identification of the near-exceedance flights, as it happened in the machine learning pipeline, can provide the analyst with relevant diagnosis and lead to operational improvements before an event occurs.

4.2. Anomaly detection in SBGR terminal airspace operations via flight tracking data analytics

For the second application, we consider an offline anomaly detection scenario in flight tracking ADS-B data. In this hypothetical situation, the goal is the post-operation discovery of anomalous operations. These may be used in air traffic controllers' training programs or aid in subsequent improvements of operational procedures with the development of new policies. The expected use case would be the daily execution of the model by domain experts followed by the evaluation of the potentially anomalous operations. As a reference, we also assume the expected number of daily flights as 500 and that the team can further investigate ten flagged anomalies in the same period.

We analyze terminal airspace arrival operations at the Sao Paulo/ Guarulhos International Airport (SBGR). Fig. 9 displays the horizontal trajectories of the analyzed flights.

For this problem, we use two model categories for anomaly detection. First, we apply an Isolation Forest model for the unsupervised discovery of potential anomalies. Then, we train an autoencoder based solely on the normal flights. By calibrating an allowable reconstruction error threshold for normal operations, the model is able to classify between normal and anomalous instances. Then, based on the identified anomalies, we apply the explanatory models based on support contextual data to assess the metrics of feature importance and logistic regression coefficients. Fig. 10 presents the conceptual summary of the framework application, while the following sections discuss the major steps in more detail.

4.2.1. Data

We analyze surveillance data capturing arrival operations in the vicinity of the Sao Paulo/Guarulhos International Airport (SBGR). The data set features 10,209 flights between December 31, 2019, and January 31, 2020.

The raw data set presents the flight information in a time series, featuring the following parameters along each flight:

- Record timestamp;
- Latitude;
- Longitude;
- Altitude;
- Heading;
- Speed;
- Aircraft model;
- Aircraft registration;
- Origin (airport);
- Destination (airport);
- Flight ID (IATA);
- Flight ID (ICAO);
- Whether the record takes place in the terminal manoeuvring area.

Finally, to emulate the model usage after training, we partition the data set temporally on the day that resulted in an approximately 80-20 split. We used the first split for training the anomaly detection models and the final 20% for replicating the model usage during real operations.

4.2.2. Anomaly detection model

Data wrangling and processing. Model conception starts with the specification of a shared data-wrangling phase, common to both models. In this case study, we analyze approach procedures in SBGR terminal airspace, from 10,000 ft onwards. To ensure data consistency with the desired flight phase and scale, the wrangling processing subjects the flights through three filters. First, it narrows down the trajectories to the subsets within terminal airspace. Second, we discard all observations previous to the first instant a given flight reached 10,000 ft. To avoid climbing flights following a takeoff captured by the surveillance equipment, we also ensure a minimum value for the rate of descent. Finally, to avoid flights with few samples, we keep only the flights with more than five observations. The result of the wrangling phase is a set of flights in the desired scale and context for the development of the models.

Unsupervised learning: Isolation forest. For the preliminary anomaly discovery phase, we construct an unsupervised model using an Isolation Forest, or iForest. The goal is to obtain an initial distinction between normal and anomalous flights. Based on this categorization, the training process of the supervised model then considers only the flights labeled as normal.

Specific preprocessing For the application of the Isolation Forest model, we calculate summarizing metrics instead of directly using the wrangled time-series data for model building. There is thus an extra step comprising the transformation of the wrangled time-series data into the tabular metrics. For each flight, we obtain the following flight-wise summarizing metrics:

- Specific Total Energy (STE)'s total, average, and standard deviation;
- Specific Potential Energy Rate (SPER)'s total, average, and standard deviation;
- Specific Kinetic Energy (SKE)'s total, average, and standard deviation;

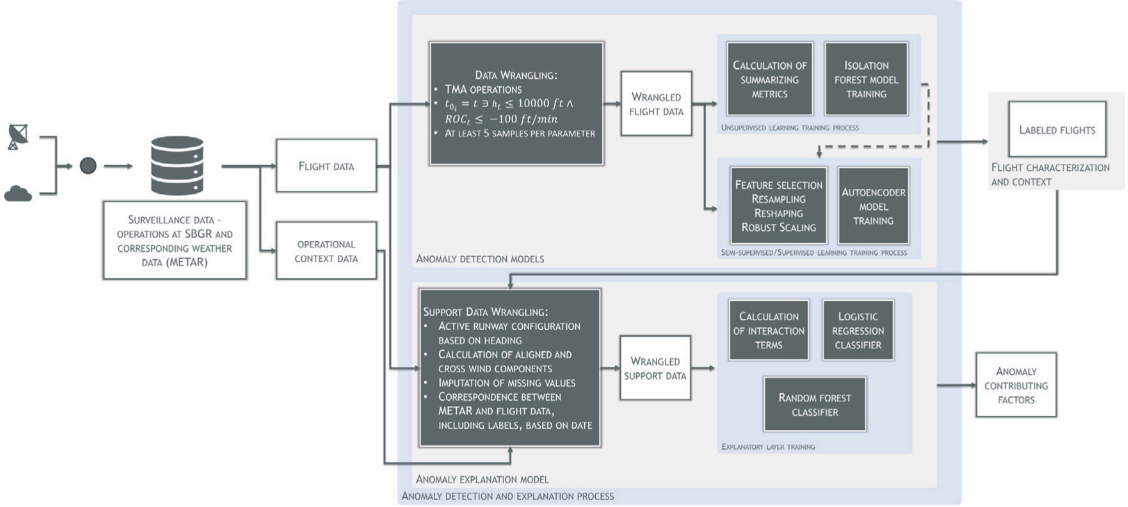
Surveillance data - operations at SBGR and corresponding weather data (METAR)
Flight data
Operational context data
Data Wrangling:
• TMA operations
• $t_{0_i} = t \ni h_t \leq 10000\ ft \wedge ROC_t \leq -100\ ft/min$
• At least 5 samples per parameter
Wrangled flight data
Calculation of summarizing metrics
Isolation forest model training
Unsupervised learning training process
Feature selection
Resampling
Reshaping
Robust Scaling
Autoencoder model training
Semi-supervised/Supervised learning training process
Anomaly detection models
Labeled flights
Flight characterization and context
Support Data Wrangling:
• Active runway configuration based on heading
• Calculation of aligned and cross wind components
• Imputation of missing values
• Correspondence between METAR and flight data, including labels, based on date
Wrangled support data
Calculation of interaction terms
Logistic regression classifier
Random forest classifier
Explanatory layer training
Anomaly contributing factors
Anomaly explanation model
Anomaly detection and explanation process

Fig. 10. Applied anomaly detection framework: SBGR terminal airspace arrival operations.

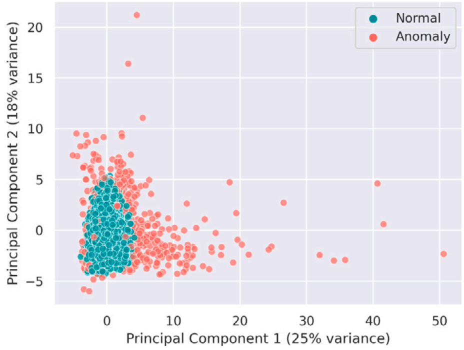

Fig. 11. Principal components, hued by the operation category (normal/anomalous).

- Total time in terminal airspace;
- Latitude at 10 000 ft;
- Longitude at 10 000 ft;
- Last recorded latitude;
- Last recorded longitude;
- The total distance flown in terminal airspace;
- Hour of the day at the entrance in terminal airspace;
- Hour of the day at the exit of terminal airspace.

Model training The Isolation Forest trained with 500 trees identified 669 anomalies within the 8073 flights selected for model training — approximately 8.3% of the data.

Fig. 11 presents the two principal components of the evaluated data, obtained via Principal Component Analysis (PCA), hued by the operation category as identified by the Isolation Forest.

Supervised learning: autoencoder. For the supervised learning process, we trained an autoencoder classifier. Autoencoders are a particular type of neural network architecture used for learning representation of data via reconstruction. It does so by encoding the data into a different feature space – also referred to as a compressed representation – before decoding it back to the original space. The underlying assumption in this approach is that if one successfully trains a model to reconstruct normal flights only, the execution of this model in an anomalous flight would result in a higher reconstruction error, enabling the classification of the flight as an anomalous instance.

Specific preprocessing As in the iForest training, the starting point was the wrangled time-series data. Instead of constructing summarizing metrics from the data, however, as performed in the Isolation Forest model training, the goal here is to use a representation as close as possible to the sequential data. Among the parameters available, discussed in Section 4.2.1, we select those representatives of the aircraft trajectory and positioning: latitude, longitude, altitude, heading, and speed.

Nevertheless, one of the challenges of anomaly detection in the aviation domain is the varying number of samples between each flight. This property of the data often requires preprocessing steps that resample and reshape the data, which is the case for the application of the autoencoder. Therefore, to unify data dimensions, we resample the time series of each individual flight with a fixed number of observations. The next step is to transform the time series of the four parameters into a single high-dimensional vector. To achieve this, we rearrange the data interspersedly. Each flight can be represented in the high dimensional space as a vector of the following shape:

$$x = [p_{0_{t_0}}, p_{1_{t_0}}, \dots, p_{n_{t_0}}, \dots, p_{0_{t_m}}, p_{1_{t_m}}, \dots, p_{n_{t_m}}]$$

where n is the number of parameters p and m is the number of time instances t.

Finally, the data goes through a scaling process. The scaling process contributes to the proper assessment of feature importance by the autoencoder neural network structure. The data is scaled via the Robust Scaler. It centers the data by removing the median and scales the result based on a specified quantile range, making the scaling process robust to outliers. Since the autoencoder models the behavior of normal flights, the contamination of the training data set with anomalous flights becomes relevant, as discussed in Section 2.2.2. The Robust Scaler is used precisely to tackle these effects of data set contamination.

Model training The training process of the autoencoder classifier relies on two fundamental steps. The first step concerns the development of a suitable neural network architecture capable of reconstructing the data with a low reconstruction error — in this case, the squared error between the norm of the actual vector and the one predicted by the autoencoder. The second training aspect refers to the definition of a reconstruction error threshold that enables classification between normal and anomalous instances.

For training a suitable neural network architecture, we further subdivide the normal data set – as flagged by the Isolation Forest model – into training and validation subsets at an 80:20 ratio.

The autoencoder neural network architecture consists of five hidden layers of 500, 300, 2, 300, and 500 neurons, respectively. For each hidden layer, we use the Rectified Linear Unit (ReLU) activation function. The neural network model was built on top of the open-source machine library scikit-learn, written in Python (Pedregosa et al., 2011).

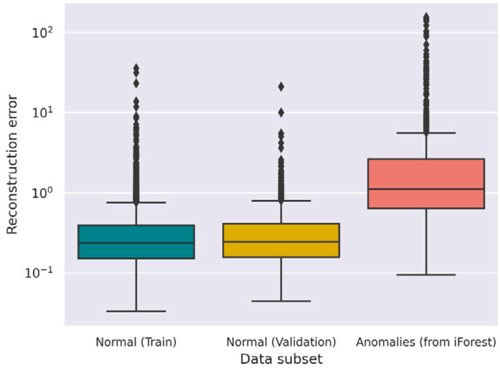

Fig. 12. Boxplot of the autoencoder reconstruction errors.

Hyperparameter tuning was not required as a simple neural network configuration sufficed for reconstructing the normal data.

Fig. 12 displays the boxplot of the autoencoder reconstruction error for the training and validation subsets, as well as for the flights flagged as anomalous by the Isolation Forest. The figure also shows that the reconstruction error distribution of the flights in the training and validation subsets are similar, indicating the model did not overfit the data. In addition, there is a statistically significant difference between the error distribution for the normal data and the ones labeled as anomalous, with higher reconstruction error values in the latter. This indicates that the model did not underfit the data and learned a distinction between normal and anomalous instances.

For defining the reconstruction error threshold, the model performance metrics are evaluated under the expected operational context. The model is expected to process 500 flights daily, and there is an analysis capacity of investigating ten flagged anomalies per day. Therefore, we assess the metrics@10, in accordance to the discussion presented in Section 3.1.5.

Performance metric evaluation happens via bootstrapping. We first calculate the reconstruction errors for observations in the training, validation, and iForest-flagged anomalies subsets. Next, we define a range of candidate thresholds. For each potential threshold value, we sample, with replacement, 500 flights and evaluate the number of flagged anomalies and the value for each performance metric. Fig. 13 shows the results when experimenting 500 times for each potential threshold. It displays the value for recall, precision, F1-Score, and false positive rate (FPR) for each threshold value. The solid lines represent the mean value for the performance metrics, while each shaded region covers the average ± one standard deviation. Additionally, the green line presents the average number of flagged anomalies per 500 flights, given a threshold.

The selected threshold value was the one that resulted in, on average, ten flagged anomalies per 500 flights - i.e., the analysis capacity in this proposed scenario. Finally, given this threshold of 3.48, we calculate the following performance metrics:

- Recall: 0.22;
- Precision: 0.84;
- F1-score: 0.34;
- False Positive Rate: < 0.01.

4.2.3. Model operation results

For simulating model operation, we use the 20% of the flights originally set apart and hence did not go through any processing step during model conception.

Out of the 2136 flights, the autoencoder classifier flagged 47 - or 2.20% - as potentially anomalous, throughout the seven days of operation. Fig. 14 displays the number of flagged anomalies per day.

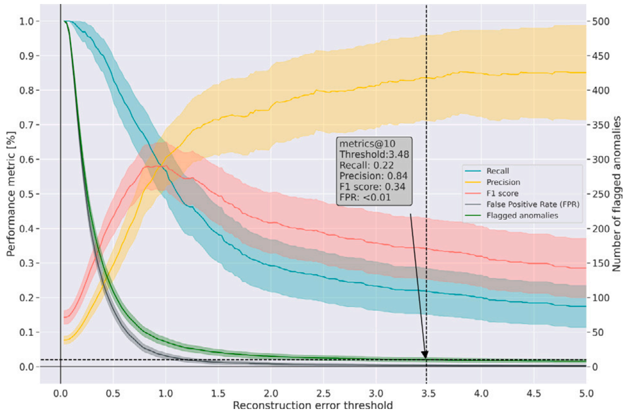

Fig. 13. Autoencoder performance metrics obtained via bootstrapping in 500 rounds, considering 500 landings per day and processing capacity of k = 10 flagged anomalies. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

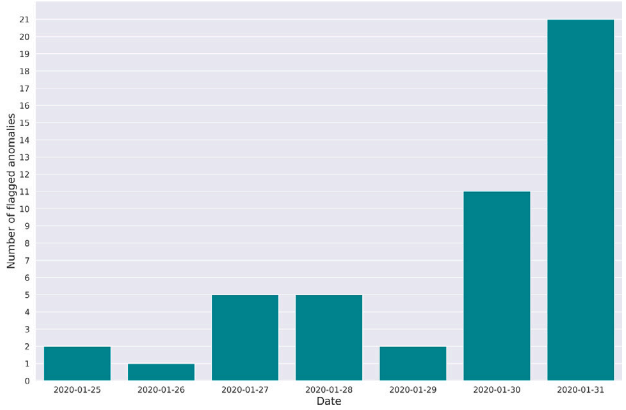

Fig. 14. Number of flagged anomalies by the autoencoder during simulated model operation.

Fig. 15 highlights the horizontal profiles of three flights flagged as potentially anomalous by the autoencoder. We see that the flights are indeed associated with operationally inefficient scenarios, presenting either a holding pattern or multiple landing attempts. This is reinforced by Fig. 16, which displays the lateral profiles of the same flights along the 95th percentile region shaded in gray.

4.3. Anomaly explanation model

Based on the class labels – anomalous or normal – obtained for each flight during model operation, our next goal is to investigate the relationship between the operational context to the identified class. For that, we build two explainable classifiers (logistic regression and random forest) that predict the anomalous quality of a flight, as per identified by the autoencoder, in accordance to operational information extracted from Meteorological Aerodrome Report (METAR) data and from the surveillance data itself. The METAR data contains information regarding winds, gusts, flight rules, presence of thunderstorms and so forth.

4.3.1. Model training

For training the explanatory models, we conducted the steps respective to the anomaly explanation model portion of the applied framework in Fig. 10.

For each flight assessed during the model operation phase – hence 20% of the original data set – we processed the corresponding METAR

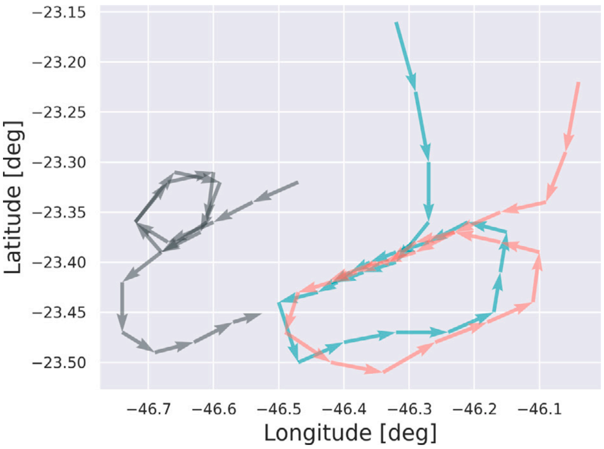

Fig. 15. Horizontal profile of three flights flagged as anomalous by the autoencoder.

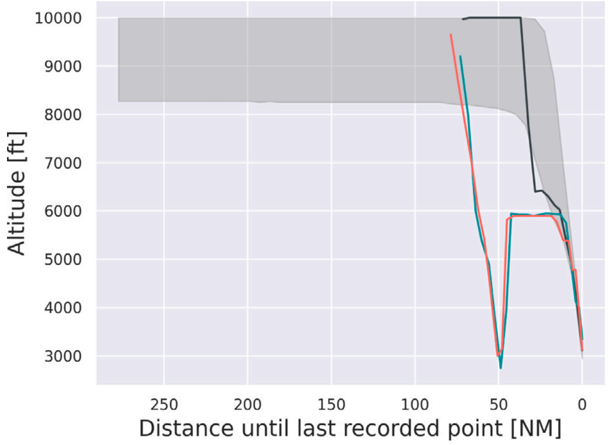

Fig. 16. Lateral profile of three flights flagged as anomalous by the autoencoder.

information to obtain data on visibility, thunderstorms, flight rules, wind gusts, and cross and aligned wind components referenced to the runway, the latter inferred via the heading parameter on the surveillance data set. Based on the wrangled support data, we explored two supervised learning classifiers: a random forest classifier, and a logistic regression classifier. For both modeling approaches, we divide the data set into train and test subsets for validating the modeling process in terms of the model capability of predicting the anomaly class of a flight given the operational context. Because the anomaly classification problem is an imbalanced one, we weight the models referenced to the class given the heuristics proposed by King and Zeng (2001). After validating the modeling process regarding the accuracy and recall on the test subset, we fit the models on the complete data set and explore the explanatory metrics.

Random forest. For the random forest, the goal is to assess feature importance to understand the operational factors contributing to the model predictions. We construct a random forest classifier with 500 trees using a split rule based on the Gini impurity, maximum depth of 15 and 3 as the number of predictors randomly sampled as split candidates. Then, we compute the importance metrics – in terms of mean decrease in impurity for each feature – and their standard deviations based on the values of each tree within the forest. Table 4 displays the confusion matrix for the random forest classifier, with an accuracy of 0.86 and a recall of 0.75.

Table 4
Random forest classifier confusion matrix.

		Prediction		
		Normal	**Anomalous**	**Total**
Actual	**Normal**	1800	289	2089
	Anomalous	12	35	47
	Total	1812	324	

Table 5
Logistic regression classifier confusion matrix.

		Prediction		
		Normal	**Anomalous**	**Total**
Actual	**Normal**	1481	608	2089
	Anomalous	18	29	47
	Total	1499	637	

Logistic regression. For the logistic regression classifier, we first augment the wrangled support data with features regarding the interaction terms between each operational factor. After fitting the model, we then analyze the β coefficients to assess the changes in the odds ratio for each operational term. Table 5 displays the confusion matrix for the logistic regression classifier, with an accuracy of 0.71 and a recall of 0.62.

4.3.2. Explanatory model results

For the random forest classifier results, we evaluate the feature importance, shown in Fig. 17. The cross-component of the wind, the headwind, and tailwind values are the top three features contributing to the predictions of the model beyond the reference dashed line of equally contributing features.

After evaluating the feature importances, we analyze the logistic regression model coefficients in terms of percent change in the odds ratio, as shown in Fig. 18. According to the values, we learn that the anomalous situations are associated with landing operations on runway 27 under wind scenarios, with an increase in the odds ratio of 62% and 58% for tailwinds and headwinds, respectively. In addition, we also see a positive association between anomalous scenarios and wind gusts and also thunderstorms accompanied by wind gusts. On the other hand, flights under Instrument Flight Rules (IFR) and increased visibility with flights under IFR showed a reduction in the odds ratio.

The constructed pipeline demonstrates its applicability regarding the case study initial objective of post-operation detection of anomalous operations as well as model explanation for discovery of contributing factors to anomalies. From the operational perspective, the results may be used in training programs or aid in subsequent developments of operational procedures. The pipeline enables efficient highlighting of

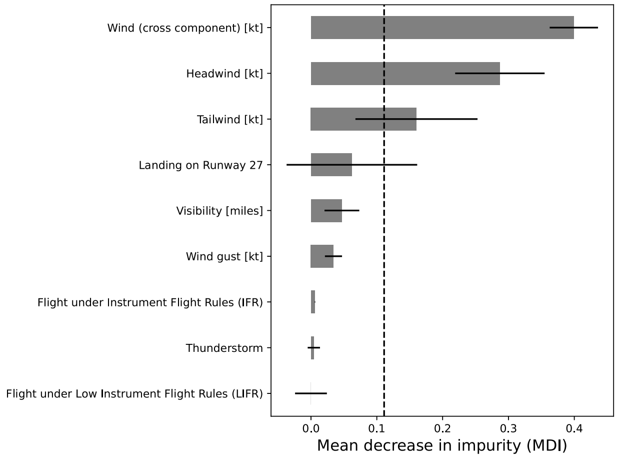

Fig. 17. Random forest classifier feature importances.

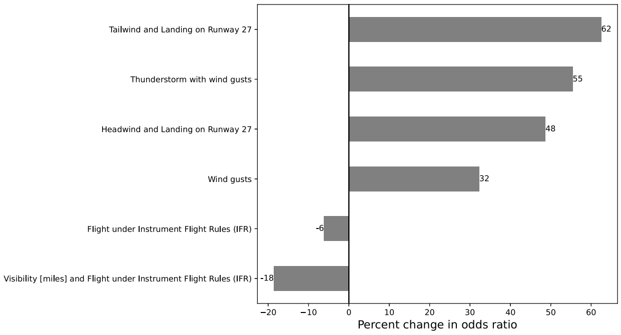

Fig. 18. Percent change in the odds ratio for selected parameters of the logistic regression model.

flights that deviate from normality. In addition, it does so without resorting to a previous mapping of what to look for: operationally inefficient scenarios surfaced without the need to resort to heuristic rules that specify every possible case. This approach can also aid the shift towards a machine learning first or AI-first philosophy, if that is an organizational goal, as models are retrained and advance in conjunction with domain expert analysis.

5. Conclusion

In this paper, we explored the anomaly detection problem in flight operations data. We discussed the fundamental aspects of the anomaly detection problem before examining it within the flight operations domain. Based on the need for a comprehensive and reusable pipeline for model building, application, and explanation, we proposed a flight data analytics framework for anomaly detection in flight operations. The framework relies on the steps of data streaming or loading and storage, model conception, and explanation while building on domain-expert analysis. The solution applies to both online and offline regimes, and at various scales of the data. We also discussed aspects of model training and strategies for model selection and operationalization. In addition, even though we focus on the flight operations domain, the framework is also applicable to analogous data mining initiatives such as flight data analysis within the maintenance domain.

We demonstrate the applicability of the framework in two scenarios of routine airline and ATM operations monitoring. In the first one, we used real aircraft performance data for landing operations at KMSP, within an unsupervised learning approach with DBSCAN. The unsupervised model was able to identify operationally significant anomalies beyond current practice. Moreover, it did so without the previous specification of the events of interest nor considering heuristics and exceedance detection rules. In addition, the pipeline identified a manageable number of potentially anomalous flights without causing work overload. In the second case study, we analyzed flight tracking data for SBGR terminal airspace arrival operations and developed an autoencoder classifier for offline anomaly detection. The supervised learning model also identified operationally significant anomalies. Furthermore, the construction of explanatory logistic regression and random forest classification models enabled the association of anomalous situations with operational factors. For instance, we learned that the anomalous situations are more likely to be associated with landing operations on runway 27 under wind scenarios, with an increase in the odds ratio of 62% and 58% for tailwinds and headwinds, respectively.

For future work, a further natural step is the application of the framework to different problem formulations with several machine learning algorithms. There are plenty of opportunities for the application of the proposed framework: it can be applied, e.g., to an online anomaly detection problem with the usage of flight tracking data, or for anomaly precursor detection with aircraft sensor data and associated FOQA events. Nevertheless, given the reusable and customizable pipeline, it can be applied to study more novel or less explored approaches for anomaly detection.

Another future research direction is to leverage this proposed framework to identify, correlate and investigate anomalous that manifest themselves in multiple scales of the flight, rather than in a single flight phase, for example.

Finally, there is a need for extending tools that support the evaluation of potentially anomalous flight, such as the explanatory approach provided in this framework, with the assessment of the indication correctness while aiding the discovery process of unknown hazards in the data. In this sense, the need for working towards further explainable AI methods applied to anomaly detection in the aviation domain continues. Together, these research topics have the potential to contribute to the effective shift towards proactive safety management and improve operational efficiency, aiding in the development of new policies by enhancing the current practice of anomaly detection in flight operations.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

Agarwala, A., Das, A., Juba, B., Panigrahy, R., Sharan, V., Wang, X., Zhang, Q., 2021. One Network Fits All? Modular versus Monolithic Task Formulations in Neural Networks. URL: http://arxiv.org/abs/2103.15261, arXiv:2103.15261 [Cs, Stat].

Alqahtani, H., Sarker, I.H., Kalim, A., Minhaz Hossain, S.M., Ikhlaq, S., Hossain, S., 2020. Cyber Intrusion Detection Using Machine Learning Classification Techniques. In: Chaubey, N., Parikh, S., Amin, K. (Eds.), Computing Science, Communication and Security. In: Communications in Computer and Information Science, Springer, Singapore, pp. 121–131. http://dx.doi.org/10.1007/978-981-15-6648-6_10.

Amidan, B.G., Ferryman, T.A., 2000. APMS SVD Methodology and Implementation. Technical Report PNWD-3026, Pacific Northwest National Laboratory, Richland, Washington, p. 21, URL: https://doi.org/10.2172/753847.

Amidan, B., Ferryman, T., 2005. Atypical event and typical pattern detection within complex systems. In: 2005 IEEE Aerospace Conference. IEEE, Big Sky, MT, USA, pp. 3620–3631. http://dx.doi.org/10.1109/AERO.2005.1559667, URL: http://ieeexplore.ieee.org/document/1559667/.

Chandola, V., Banerjee, A., Kumar, V., 2009. Anomaly detection: A survey. ACM Comput. Surv. 41 (3), 58. http://dx.doi.org/10.1145/1541880.1541882, URL: http://doi.acm.org/10.1145/1541880.1541882.

Das, S., Matthews, B.L., Srivastava, A.N., Oza, N.C., 2010. Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study. In: KDD '10: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 47–56, doi: https://dl.acm.org/doi/10.1145/1835804.1835813.

Departamento de Controle Do Espaço Aéreo (DECEA), 2012. Plano de Implementação. ATM Nacional, URL: https://publicacoes.decea.mil.br/publicacao/pca-351-3.

Deshmukh, R., 2020. Data-Driven Anomaly and Precursor Detection in Metroplex Airspace Operations (Doctorate). Purdue University, West Lafayette, Indiana, URL: https://doi.org/10.25394/PGS.12121095.v1, Artwork Size: 13674965 Bytes Publisher: Purdue University Graduate School.

Deshmukh, R., Hwang, I., 2019. Anomaly Detection Using Temporal Logic Based Learning for Terminal Airspace Operations. In: AIAA Scitech 2019 Forum. American Institute of Aeronautics and Astronautics, San Diego, California, http://dx.doi.org/10.2514/6.2019-0682, URL: https://arc.aiaa.org/doi/10.2514/6.2019-0682.

Deshmukh, R., Sun, D., Hwang, I., 2019. Data-Driven Precursor Detection Algorithm for Terminal Airspace Operations. In: Thirteenth USA/Europe Air Traffic Management Research and Development Seminar (ATM2019). p. 7.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In: KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining. KDD '96, pp. 226–231, doi: dl.acm.org/doi/10.5555/3001460.3001507.

European Commission, 2011. Flightpath 2050: Europe's Vision for Aviation. Technical Report, European Union, Luxembourg.

FAA, 2004. AC 120-82 Flight Operational Quality Assurance.

FAA, 2019. NextGen Implementation Plan 2018–19.

Federal Aviation Regulations, 2009. 14 CFR 135.152 – Flight data recorders. URL: https://ecfr.federalregister.gov/current/title-14/chapter-I/subchapter-G/part-135/subpart-C/section-135.152.

Fernandez, A., Martınez, D., Hernandez, P., Cristobal, S., Schwaiger, F., Nunez, J.M., Ruiz, J.M., 2019. Flight Data Monitoring (FDM) Unknown Hazards detection during Approach Phase using Clustering Techniques and AutoEncoders. p. 8.

Gorinevsky, D., Matthews, B., Martin, R., 2012. Aircraft anomaly detection using performance models trained on fleet data. In: 2012 Conference on Intelligent Data Understanding. IEEE, Boulder, CO, USA, pp. 17–23. http://dx.doi.org/10.1109/CIDU.2012.6382196, URL: http://ieeexplore.ieee.org/document/6382196/.

Hariri, S., Kind, M.C., Brunner, R.J., 2021. Extended Isolation Forest. IEEE Trans. Knowl. Data Eng. 33 (4), 1479–1489. http://dx.doi.org/10.1109/TKDE.2019.2947676, URL: https://ieeexplore.ieee.org/document/8888179/.

Höhndorf, L., 2018. Statistical Dependence Analyses of Flight Data for Safety Management (Ph.D. thesis). Technische Universität München.

ICAO (Ed.), 2013. Doc 9859, Safety Management Manual (SMM), third ed. In: Doc / International Civil Aviation Organization, vol. 9859, ICAO, Montreal, OCLC: 931319110.

ICAO, 2019. Doc 10004, Global Aviation Safety Plan, 2020 - 2022 ed. ICAO, Montreal.

IHLG, 2019. Aviation Benefits Report. Technical Report, Industry High Level Group, p. 76.

King, G., Zeng, L., 2001. Logistic Regression in Rare Events Data. p. 27.

Li, L., Hansman, R.J., 2013. Anomaly Detection in Airline Routine Operations Using Flight Data Recorder Data (Ph.D. thesis). Massachusetts Institute of Technology, Cambridge, MA.

Liu, F.T., Ting, K.M., Zhou, Z.-H., 2008. Isolation Forest. In: 2008 Eighth IEEE International Conference on Data Mining. IEEE, Pisa, Italy, pp. 413–422. http://dx.doi.org/10.1109/ICDM.2008.17, URL: http://ieeexplore.ieee.org/document/4781136/.

Liu, F.T., Ting, K.M., Zhou, Z.-H., 2012. Isolation-Based Anomaly Detection. ACM Trans. Knowl. Discov. Data 6 (1), 1–39. http://dx.doi.org/10.1145/2133360.2133363, URL: https://dl.acm.org/doi/10.1145/2133360.2133363.

Lochner, M., Bassett, B.A., 2021. Astronomaly: Personalised active anomaly detection in astronomical data. Astron. Comput. 36, 100481. http://dx.doi.org/10.1016/j.ascom.2021.100481, URL: https://www.sciencedirect.com/science/article/pii/S2213133721000354.

Lundberg, S.M., Lee, S.-I., 2017. A unified approach to interpreting model predictions. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (Eds.), Advances in Neural Information Processing Systems, Vol. 30. Curran Associates, Inc., URL: https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

Matthews, B., Srivsatava, A.N., Schade, J., Schleicher, D.R., Chan, K., Gutterud, R., Kiniry, M., 2013. Discovery of Abnormal Flight Patterns in Flight Track Data. In: 2013 Aviation Technology, Integration, and Operations Conference. American Institute of Aeronautics and Astronautics, Los Angeles, CA, http://dx.doi.org/10.2514/6.2013-4386, URL: http://arc.aiaa.org/doi/10.2514/6.2013-4386.

Murça, M.C.R., 2018. Data-driven modeling of air traffic flows for advanced Air Traffic Management (Ph.D. thesis). Massachusetts Institute of Technology, URL: https://dspace.mit.edu/handle/1721.1/120378, Accepted: 2019-02-14T15:22:40Z.

NASA's Discovery in Aeronautics Systems Health (DASHlink), 2012. Sample Flight Data. URL: https://c3.nasa.gov/dashlink/projects/85/.

Olive, X., Basora, L., 2019. Identifying Anomalies in past en-route Trajectories with Clustering and Anomaly Detection Methods. p. 11.

Oster, C.V., Strong, J.S., Zorn, C.K., 2013. Analyzing aviation safety: Problems, challenges, opportunities. Res. Transp. Econ. 43 (1), 148–164. http://dx.doi.org/10.1016/j.retrec.2012.12.001, URL: https://linkinghub.elsevier.com/retrieve/pii/S0739885912002053.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine learning in python. J. Mach. Learn. Res. 12, 2825–2830.

Rahmah, N., Sitanggang, I.S., 2016. Determination of Optimal Epsilon (Eps) Value on DBSCAN Algorithm to Clustering Data on Peatland Hotspots in Sumatra. IOP Conf. Ser.: Earth Environ. Sci. 31, 012012. http://dx.doi.org/10.1088/1755-1315/31/1/012012, URL: https://iopscience.iop.org/article/10.1088/1755-1315/31/1/012012.

Rousseeuw, P.J., 1987. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. 20, 53–65. http://dx.doi.org/10.1016/0377-0427(87)90125-7, URL: https://linkinghub.elsevier.com/retrieve/pii/0377042787901257.

Ruff, L., Kauffmann, J.R., Vandermeulen, R.A., Montavon, G., Samek, W., Kloft, M., Dietterich, T.G., Muller, K.-R., 2021. A Unifying Review of Deep and Shallow Anomaly Detection. Proc. IEEE 109 (5), 756–795. http://dx.doi.org/10.1109/JPROC.2021.3052449, URL: https://ieeexplore.ieee.org/document/9347460/.

SESAR Joint Undertaking, 2019. European ATM master plan: digitalising Europe's aviation infrastructure : executive view : 2020 edition. Publications Office, LU, URL: https://data.europa.eu/doi/10.2829/10044.

Wei, Q., Ren, Y., Hou, R., Shi, B., Lo, J.Y., Carin, L., 2018. Anomaly detection for medical images based on a one-class classification. In: Medical Imaging 2018: Computer-Aided Diagnosis, Vol. 10575. International Society for Optics and Photonics, p. 105751M. http://dx.doi.org/10.1117/12.2293408, URL: https://www.spiedigitallibrary.org/conference-proceedings-of-spie/10575/105751M/Anomaly-detection-for-medical-images-based-on-a-one-class/10.1117/12.2293408.short.