Safety Science 114 (2019) 89–104

Contents lists available at ScienceDirect

Safety Science

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

Using machine learning methods in airline flight data monitoring to generate new operational safety knowledge from existing data

Julian Oehling, David J. Barry*

Safety & Accident Investigation Centre, Cranfield University, Cranfield, Beds MK43 0TR, United Kingdom

ABSTRACT

The aim of this work is to investigate the possibility of using machine learning (ML) methods in order to generate novel, safety-relevant knowledge from existing flight data. Airlines routinely generate vast amounts of flight data from routine monitoring, but the concept of extracting safety knowledge from this data is still based on detecting exceedances of expert-defined thresholds. This system is conceptually unable to detect novel occurrences for which no such filters exist. ML techniques are able to close this gap.

This paper first reviews the literature to select an appropriate ML method. A form of unsupervised learning called “Local Outlier Probability” is selected. Next, an appropriate feature space is developed and implemented in the flight data monitoring system of a supporting airline to generate the dataset. This dataset is cleaned and the outlier calculation performed. The results are statistically analysed. Furthermore, the top outliers are reviewed by the airline’s review pilots in the same way as the traditional exceedance events. Last, the severities and safety relevance of both types of events are compared.

This work successfully shows that the chosen approach is able to reduce the number of undetected safety-relevant occurrences by finding novel occurrence types which were undetected by a contemporary and mature flight data monitoring system.

This research builds on recent literature by developing a novel method which can be scaled to work in an airline production environment with large datasets, as demonstrated by the efficient analysis of 1.2 million flights.

1. Introduction and problem formulation

A vast amount of flight data recorded onboard aircraft is created each day. Major airline groups conduct hundreds of thousands or even millions of flights each year (American Airlines, 2016; United Airlines, 2016). Each flight produces on average several hours of data, in which thousands of parameters can be recorded between one and eight times per second. Airlines and other aircraft operators are required to monitor this data with the purpose of improving flight safety (EASA, n.d.). Generally, airlines are interested in flights during which abnormal data patterns were recorded. Even though the concept of Safety II (Hollnagel, 2014) highlights the importance of using the normal data as well, the emphasis of flight data monitoring (FDM) is to detect abnormal occurrences which did not, but could have, credibly escalated into an accident.

Due to the amount of data, it is impossible to use human experts to review all recorded data. Instead, the concept of FDM relies on a system which highlights relevant flights or portions of flights. A portion of a flight flagged by the system as potentially interesting is commonly referred to as an “event”. In order to generate these events, exceedance detection algorithms are the state of the art approach. This concept of checking the flight data against pre-determined threshold values, which were set by subject matter experts, and flagging flights in which one threshold or a combination thereof were exceeded, has been used for decades (Federal Aviation Administration, 2004). During this time, it has been continually improved and fine-tuned and is now working well and trusted by the industry. However, there are two major drawbacks:

> First, if a threshold is incorrectly set, this can normally only be detected if the error results in too many events, so-called false-positives. In this case, these false events will be investigated and the cause can be found and corrected. If, however, the threshold is set too wide and thereby creating too few or no events at all – called false-negatives – then there is a high probability that this will not be noticed since the data without events is rarely reviewed.
> Second, events can only be triggered for occurrences for which an event was designed. In other words, there may be false-negatives in the data because no one has yet imagined such an incident to occur.

A typical event set may comprise over one hundred algorithms (Civil Aviation Authority, 2013) to detect occurrences such as high speeds below specified altitudes, exceedances of flight manual limits and exceedances of normal pitch and roll attitudes. Whilst the typical event sets are quite comprehensive, it is very possible to have an abnormal flight which is not detected as such by the predefined algorithms.

[*] Corresponding author.
E-mail addresses: JulianOehling@eMail.de (J. Oehling), d.j.barry@cranfield.ac.uk (D.J. Barry).

https://doi.org/10.1016/j.ssci.2018.12.018
Received 23 May 2018; Received in revised form 18 December 2018; Accepted 19 December 2018

These traditional event sets are rooted in a time when it was difficult to handle the flow of data from aircraft, particularly if the airline had a large fleet, however the methods to handle large amounts of data have been intensively researched and improved in recent years. The field of machine learning (ML) has developed conceptually new approaches to use computationally intensive methods to extract hidden knowledge from data (Marr, 2015).

The aim of this work is to use ML tools in order to increase the detection of unknown occurrences (false negatives) in real-world airline data. The detection of previously unknown occurrences should lead to improved hazard identification and enhanced risk mitigation.

The aim is not to replace existing methods, but to introduce a complementary method to allow enhanced safety knowledge discovery.

The resulting system must fulfil the following requirements:

- Detect unknown occurrences: The system should not rely on pre-programmed definitions but use the entire available data to find safety-relevant events. It is not the goal to imitate an existing exceedance monitoring system, but to complement it by finding otherwise undetected false negatives
- Handle large amounts of data: The system should be able to work with millions of flights and deliver results within 2 or 3 days (a timescale likely to be acceptable to an operational safety department)
- Handle diverse data: The operation of a large airline will likely cover multiple aircraft types and many different airports. The system should not be restricted to a limited set of airports or aircraft
- Deliver useful results: The results produced should not only be of academic interest but highlight safety-relevant occurrences which would otherwise be undetected

To achieve these goals, the following steps are undertaken: Through literature review, the current state of the art in ML and ML in FDM is established and an appropriate ML concept chosen. Thereafter, a dataset is generated from the FDM system of a supporting airline. This dataset is then verified, cleaned and pre-processed as necessary. In the next step, the ML tool is used to highlight interesting flights. Then the results are compared to the existing exceedance detection system and the most relevant new findings are reviewed by subject matter experts. Lastly, the overall usefulness of the new system is discussed and further possibilities for improvement are suggested.

Note that the term "flight data" used here refers specifically to data recorded by airborne flight data recorders known as quick-access recorders (QAR) carried on aircraft, rather than a generic reference to data about flights. "Flight data", as used here, does not refer to air traffic control data (as used by West and McCluskey (2000) for example), occurrence reporting or maintenance data.

2. Literature review

The approach used in this work uses machine learning algorithms on data from FDM systems. Both FDM and machine learning have had little overlap in the past, hence this literature review first covers machine learning on its own before looking at previous work on FDM machine learning in the second part.

2.1. Machine learning

The term ML generally refers to algorithms which learn from data. This can be understood as a software which builds a model based on the input data rather than working with a predefined model which was encoded into the software during the development of the algorithm. Mitchell (1997) concludes that ML is achieved when a computer program's performance increases with relevant experience.

The terms ML and "data mining" are often used interchangeably. Some authors define ML as focused more on prediction and data mining as focused on discovery, but especially in the subfield of unsupervised learning, these definitions are often mixed.

ML is broadly divided into the subcategories of supervised and unsupervised learning. The main difference between these two is that in *supervised* learning a correct answer for a learning set of data is already known. For example, with a set of handwriting samples and existing correct transcriptions into digital text a machine can be trained to recognise new handwriting samples. In *unsupervised* learning, there is no known solution which can be used to train the algorithm. Unsupervised learning is commonly used to structure large datasets by means of clustering, or, as in this paper, to find outlier data in a large data pool. While there are also some examples of supervised outlier detection methods, the common approach is to treat outlier detection as an unsupervised ML task.

2.1.1. Outlier detection

Outlier detection is used to classify unusual, novel or anomalous observations in data. Its use is therefore widespread and Hodge and Austin (2004) list a number of examples, including:

- Fraud detection in banking and other systems
- Detecting novelties in images
- Time series monitoring
- Fault diagnosis

They explain that outlier detection is often used in safety-critical domains for monitoring degradation in mechanical systems. Mechanical systems are well-suited to outlier detection because the system has boundaries in terms of operation and behaviour. In other words, there are a finite number of measures which could adequately describe the functioning of the system. There are fewer constraints and more potential for variability in non-mechanical systems and processes, such as an airliner flying from A to B, however, while the system might be more dynamic, constraints may still exist in the form of standard operating procedures or regulatory requirements. Therefore, meaningful measures can be taken to describe the performance of parts of the process, which can then be used for outlier detection.

In order to detect outliers from a dataset several concepts are commonly used, including:

- Extreme value analysis
- Probabilistic and statistical model-based approaches
- Proximity-based approaches
- Angle-based approaches
- Artificial neural networks

A brief summary of some of these concepts is provided below, however the reader is directed to more comprehensive reviews such as Hodge and Austin (2004), Markou and Singh (2003a, 2003b) and Pimentel et al. (2014).

2.1.1.1. Extreme value analysis. The simplest approach to outlier detection is to look at the extreme values in a given set of observations. A threshold can be set above or below which represents an outlier. The threshold may be set by, for example, assuming the data is normally distributed and selecting a threshold a number of standard deviations away from the mean. Laurikkala et al. (2000) use simple box plots to produce a graphical representation of the data so that outliers can be readily identified. Extreme value analysis is mostly suited to one-dimensional data.

2.1.1.2. Probabilistic and statistical model-based approaches. In model-based approaches, the concept is to fit a model to the data and search for data points which are not explained by this model. Examples for such models are common statistical distributions (Gauss, Poisson etc.) or convex hulls fitted around the data points in two- or three-dimensional space (support vector machines – see Tax et al. (1999) for an example). Another variant of this approach is to search for points which minimise the variance of the given dataset when they are excluded. The underlying idea is to find the outliers at the outer edge of the data space. These approaches tend to work better when the underlying data distribution is known as they assume the distributions are Gaussian. Markou and Singh (2003a) provide a thorough review of statistical model-based approaches.

2.1.1.3. Proximity-based approaches. In proximity-based approaches, the governing factor to determine whether a data point is considered an outlier or not is either the distance to its neighbour points or the density of other points around the point in question. Distance-based approaches can be understood as defining a radius around the point and counting the number of other data points that fall within this radius. If the result is below a pre-defined threshold, the data point is considered an outlier. Similarly, the inverse of this number can be considered as an outlier score, where a high number of neighbouring data points creates a low outlier score and vice versa. This approach tends to perform generally well and different variations are commonly used, for example, k-Nearest Neighbours (kNN) and Orca. One of the disadvantages however, is their decreasing performance when the data density is of high variability. In this case, the use of a density-based algorithm is preferable. Density-based approaches are based on the concept that the density of other points around an outlier should be significantly different from the density around its neighbours. One common implementation of this concept is the Local Outlier Factor (LOF). Breunig et al. (2000) developed the concept for this algorithm which uses local densities to determine outliers.

Fig. 1 visualises a dataset with an area of lower density (C1) and one of higher density (C2). It further contains two outliers, O1 and O2. While a distance-based approach should be able to detect data point O1 as an outlier, since the distances to its neighbours are unusually high compared to the average, only a local density-based approach can detect outlier O2. It is not further away from its neighbouring points than an average point in C1, but the density around it is less than the local density around its neighbours in C2. The LOF algorithm returns a dimensionless number as result for each point, which describes the outlierness. A value significantly greater than 1 generally characterizes an outlier. The runtime and memory requirements of LOF algorithms scale by n2, which means that a dataset of twice the size requires four-times as much memory and calculation time. Some variants of LOF specifically aim at decreasing the computation time by either reducing the time spent on the calculation of LOF scores for data points deep within clusters (FastLOF) or improving its performance when using multiple computing cores in parallel (Goldstein, 2012).

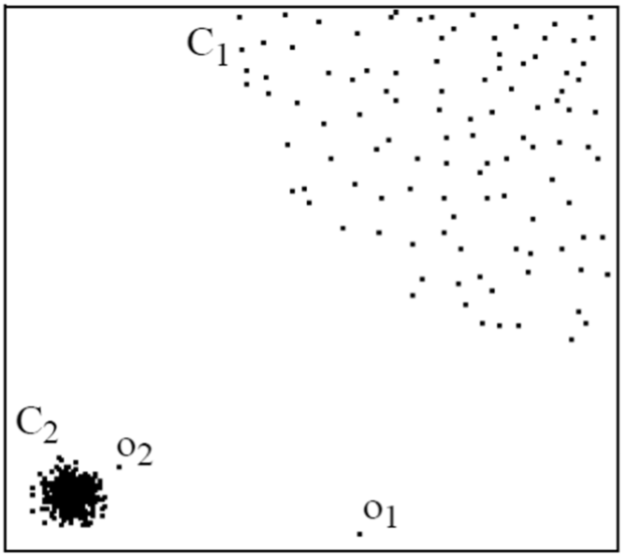

Fig. 1. Local Outlier Factor concept.
Source: Breunig et al., 2000, p. 94.

Lazarevic et al. (2003) found that LOF compared well to other approaches when trying to find novel network intrusion events. Network attacks are characterised by relatively sparse outliers amongst large amounts of routine data, which is similar to safety events in FDM data. Campos et al. (2016) evaluated different outlier detection approaches and found that classic LOF still remained amongst the “state-of-the-art” approaches.

One improvement with regards to the interpretability of the LOF score is the Local Outlier Probability (LoOP) algorithm (Kriegel et al., 2009). The LoOP algorithm uses a concept of probabilistic distances of the nearest neighbours to estimate local densities. The density around a sample is compared to the density around its neighbours and the normalized difference serves as a measure of outlierness. It returns a probability in the range between 0 and 1, which is directly interpretable as the probability that this data point is an outlier. Due to this desirable feature as well as all the advantages of the LOF, the LoOP is used in this work as the primary approach to detect outliers in flight data. The Kriegel algorithm has been used in this paper without modification, therefore the reader is directed to Kriegel et al. (2009) for a full description.

2.1.1.4. Angle-based approaches. Angle-based approaches use the range of angles from one data point to its neighbours as a measure of its outlierness (Kriegel et al., 2008). The basic idea is that a data point in the centre of a cluster will have its neighbours approximately evenly spread around in all directions. Therefore, the angles between an arbitrary reference line and the line which connects the data point to a sufficient number of neighbours should fall within a wide range, i.e. 360°. For an outlier, however, the range should be much narrower. In Fig. 1 the outlier O2 has all its neighbours on its lower left side and all angles are within a range of approximately 90°. The smaller the range of angles, the higher the probability of the point in question being an outlier. In theory, this approach is very robust to the increase in dimensionality, because angular based calculations are very stable when the number of dimensions increases. This is an advantage over proximity-based methods, where with increasing dimensionality the distances between two points tend to increase and the data is therefore becoming more sparsely distributed, a problem often called “curse of dimensionality”. One implementation is the Angle-Based Outlier Detection (ABOD). The main disadvantage of this approach is the required processing time. The algorithms scale at n^3, which means that a dataset twice as big requires an eight-fold increase in processing time and memory usage. This makes the algorithm unsuitable for large datasets like the one used in this paper. Some improved implementations (LB-ABOD, FastABOD) have been developed to reduce processing time, however even though they are faster for a given dataset, they scale at n^3 as well, making them unsuitable for very large datasets.

2.1.1.5. Artificial neural networks. Like the biological neural networks they imitate, artificial neural networks (ANN) learn by example and can be trained to classify observations (e.g. as outliers). They are very powerful and capable of handling high-dimensional data, making them suitable for classification in the realm of flight data monitoring, as demonstrated by Nanduri and Sherry (2016).

2.1.2. Machine learning in flight data monitoring

Research in the area of FDM has been limited in the past. This may be due to several factors. For one, the access to the data is very limited. The relevant European regulation (EASA, n.d.) requires the operator to adequately protect the data. Since flight data is only recorded by the operator and may contain information about events which could be harmful to the operator's reputation, there is very little interest to share this data for academic research. Furthermore, the degree of subject matter expertise required to understand flight data is high and the number of practitioners in this field is relatively low, especially compared to fields such as business administration or information technology, which are common areas of ML application. Therefore, the combination of limited data and limited subject matter expertise available results in few publications on ML in FDM.

The literature that does exist has tended to use between hundreds and tens-of-thousands of flights (e.g. Jesse, 2011; Mendes, 2012; Li and Hansman, 2013), and often focussed on a single aircraft type (fleet). This research differs in that the methods used are suitable for scaling into a production environment in a large airline, as demonstrated by the large dataset used. The methods here allow such large datasets to be analysed efficiently, which is crucial in an environment where data generation never stops. The methods used here can also handle diverse data relating to different fleets and routes, something that has not been explored in much of the research described below.

Li and Hansman (2013) have carried out the most complete research in the field. They developed two products, ClusterAD-Flight and Cluster AD-Data Sample, which cover the entire process from data transformation, dimension reduction, cluster detection, outlier detection and visualisation of the results. The underlying outlier detection technique behind these tools is Density-Based Spatial Clustering of Applications with Noise (DBSCAN), which was developed by Ester et al. (1996) and can discover both clusters and outliers from datasets even when noise is present. The developed tools were tested in several studies on datasets of up to approximately 26,000 flights and were able to detect occurrences of interest (Das et al., 2012; Li et al., 2016, 2015; Zhao et al., 2015).

The research described here follows on from that of Li and Hansman (2013) and Li et al. (2015). However this approach differs by reducing the dimensionality of the data to 60 dimensions, relating to safety specific features, which allows a much larger number of flights to be analysed (in this case 1.2 million). Limiting the choice of dimensions to generic measures, which are largely indiscriminative of aircraft type or route flown, allows comparisons to be made between flights to different airports by different fleets. For example, during an aircraft's final approach to a runway, it would be expected that there would be little *deviation* from the runway heading, but the runway heading will vary depending on the airport. Likewise, this *deviation* measure will be independent of aircraft type (fleet), thus allowing clustering methods to focus on safety specific features, rather than general differences in operating environment.

Jesse (2011) investigated the usefulness of different clustering algorithms on a dataset using two dimensions at a single time point from 100 flights by one single aircraft type during approaches to a single runway.

Mendes (2012) used the supervised learning technique called Support Vector Machines (SVM) in order to investigate 629 automatic landings as part of an autoland study by an operator. He was able to label 518 landings as normal and detected 111 outliers, which needed further investigation. By labelling more than 80% of the automatic landings as normal he could reduce the workload of the flight data analysts drastically, who would otherwise have to review every single landing.

Smart (2011) also used SVM in his analysis of 1518 flights into a single airport by a single aircraft type. He developed a suitable feature space and used metrics such as the F-score to rate the quality of the results achieved by ML techniques. Even though he was specifically looking for novelty detection, he managed to reproduce 84% of the results created by conventional FDM methods.

Biswas et al. (2013) used semi-supervised ML methods such as DBSCAN in different case studies using between 100 and approximately 2500 flights. It could be shown that in conjunction with expert knowledge these approaches did produce meaningful results, some of them clearly safety-relevant.

Matthews et al. (2013) investigated the use of different data mining algorithms to identify safety-relevant occurrences of different varieties in up to 19,243 flights. They found that in collaboration with review pilots it was possible to detect novel threats through this data mining approach.

2.1.3. General issues with machine learning

A frequent criticism of machine learning methods is that they lack transparency and that it can be difficult for the end-user to determine how any conclusion or finding was reached. In the domain considered here, that of a safety-critical system (SCS), resources cannot be squandered on false-positives nor targeted at issues which have little in the way of supporting evidence.

In another SCS, that of medicine, Kononenko (2001) assesses the relative transparency and ability to explain the performance of several ML methods and finds that proximity-based methods, such as kNN, rate quite poorly. This is echoed by Kotsiantis et al. (2006). However, this is not a significant issue in the domain of FDM, where it is normal for suspected abnormalities, identified through event algorithms, to be manually validated by an expert. Resources exist to perform this function as part of any FDM programme, due to the vagaries of flight data (sensor anomalies, environmental factors and so on). The challenge is to enhance the detection of suspect abnormalities so that the expert has the opportunity to inspect the data and decide if further investigation is required. Due to this, and the fact that the methods described here are intended to complement existing methods, the transparency and explanatory ability of ML is not considered detrimental.

3. Methods

3.1. Method choice

Extreme value analysis, as already stated, is most suited to one dimensional data, making it unsuitable for the multi-dimensional domain of flight data and the analysis of aircraft operations. Similarly, probability model-based approaches are also best suited to limited dimensions.

Proximity based methods do seem to be a suitable candidate for the problem, as they are relatively simple and can handle large volumes of data with multiple dimensions. Angle-based methods could be potential candidates, however the scalability issues are likely to be prohibitive in a production environment.

Whilst ANNs show considerable promise in the realm of flight data classification, a key aim of this research is to detect outliers in large datasets (>1 million flights), in a production environment (e.g. airline safety department, rather than computer laboratory or research institution), hence the relative simplicity of a proximity-based approach is preferred.

3.2. Method overview

A basic overview of the methods is provided in Fig. 2 below.

The data has been generated by approximately three hundred aircraft, belonging to six different fleets (e.g. A320 series) and 14 subfleets (e.g. A321 specific type), which conduct more than 1000 flights each day. They operate in a global route network to 895 different runways. For this work data of flights from March 2013 until March 2016 were available.

Each flight records between 150 and 2000 different parameters with sampling rates that normally range between 1 Hz and 8 Hz. Each parameter set is individually adapted to the specific aircraft type, however all of them have basic parameters in common, such as airspeed, heading, geographic position and altitude.

Airbus aircraft accident statistics (Airbus, 2016) show that 31% of

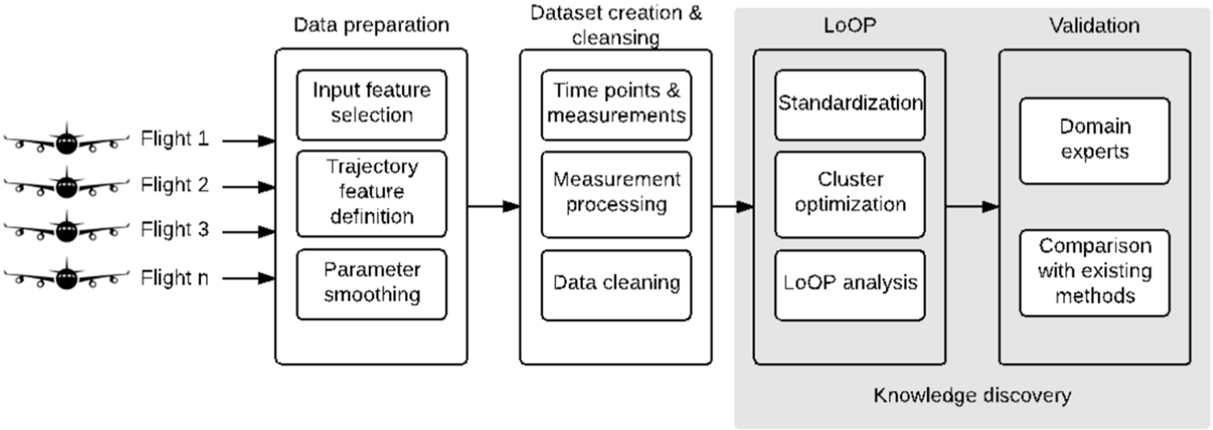

Fig. 2. Overview of methods.

all aviation accidents occur during the approach phase, more than during any other flight phase. The approach phase is the period between descent and landing and is characterised by the deployment of high-lift devices, landing gear and establishing on the final approach track. Boeing statistics (Boeing, 2016) support those of Airbus, hence this work concentrates on the approach phase of flight, or more specifically, the last 10NM before the landing.

In order to cope with the diversity of airports and aircraft, the dataset has to be defined in a way that is independent of individual technical characteristics of the aircraft or geographical position of the runway. The resulting subset is referred to as the feature space.

3.3. Development of the feature space

The aim of the feature space development is to reduce the available data to a smaller subset, which can be used to detect outliers. This subset needs to be independent of the technical characteristics of each aircraft or of the geographical position of the landing runway. Therefore, the approach phase of the flight needs to be described by the feature space in a generic and efficient way. However, the feature space must also be specific enough to describe each approach phase with adequate precision to judge its outlierness relative to other approaches.

Since the aim of this work is to detect safety-relevant occurrences in the data, the focus of the feature space is on describing the safety-relevant properties of each approach. According to the Airbus and Boeing aircraft accident statistical summaries (Airbus, 2016; Boeing, 2016), the majority of all accidents fall into just three major accident categories: Loss of control in flight (LOC-I), controlled flight into terrain (CFIT) and runway excursions (RE).

CFIT accidents can be understood as an outcome of an unsafe trajectory, where the aircraft is either at an inadequate height or an inappropriate geographical position. RE, on the other hand, are mostly caused by improper aircraft energy management, usually allowing the aircraft to carry too much energy. LOC-I accidents can be caused by multiple factors, but are often caused by allowing the airspeed to reduce below a minimum threshold and entering a stall. In order to capture these three major categories, the feature set should describe the aircraft trajectory and energy level during approach.

3.3.1. Feature space parameter summary

The measurements used to define the feature space are summarised in Table 1. All of these features can be derived from flight data recorded onboard the aircraft.

3.3.2. Trajectory description

To compare approach trajectories towards different runways, a runway-based coordinate system is developed.

This coordinate system uses the threshold as a reference point and the extended runway centreline and the runway elevation as frames of reference for the trajectory description. The basic layout and a selection of measurements are visualised in Fig. 3.

The time points $t_1, t_2, t_3, t_4 \ldots t_{10}$ are defined as the points in time when the aircraft has a ground track distance of 1, 2, 3, 4 … 10 nautical miles (NM) remaining until the threshold of the runway. The ground track distance is calculated by integrating the best available recorded ground speed parameter. The runway threshold is defined as the point at which the aircraft is at 50 ft radio altitude during the landing.

At each time point, the orthogonal distance of the aircraft from the extended runway centreline (c) is measured ($c_1, c_2, c_3, c_4 \ldots c_{10}$). The calculation of c is provided in Appendix C and, combined with the distance to the threshold, it describes the position of the aircraft over the ground relative to the runway.

The aircraft's altitude is measured as height (ft) above runway elevation (h) at each time point ($h_1, h_2, h_3, h_4 \ldots h_{10}$). This allows the comparison of approaches into airports with different field elevations. The combination of distance to threshold, centreline deviation and height over runway elevation describes the aircraft's position in three-dimensional space.

To measure the energy level, the aircraft's airspeed is measured at each time point. Any difference in approach speeds between fleets may create different clusters, however it should not cause increased outlierness.

Besides these measurements of position and energy, more measurements are created to capture the current trend for each of the previous measurements, i.e. whether the deviation is increasing or decreasing.

The parameter "i" measures the angle between the aircraft's current track (as recorded on board) and the runway orientation at each time point ($i_1, i_2, i_3, i_4 \ldots i_{10}$). If the value of this measurement is less than zero, the aircraft is flying on a course which points to the left of the approach course, if it is larger, it is flying to the right relative to the approach course. Together with the centreline deviation, this measurement indicates whether the distance between the aircraft and the extended centreline is increasing or decreasing.

The parameter "f" measures the aircraft's current flight path angle (as recorded on board) at each time point ($f_1, f_2, f_3, f_4 \ldots f_{10}$). This measurement serves as a description of the trend of the height above the landing runway.

The parameter "a" measures the acceleration along the flight path angle at each time point ($a_1, a_2, a_3, a_4 \ldots a_{10}$), and represents the trend in airspeed.

The entire feature space consists of 10 measurements taken across 6 parameters:

Table 1
Feature space summary.

Notation	Description	Measured
c	Lateral distance from runway centreline (NM)	At time point
h	Height above runway threshold (feet)	At time point
s	Average airspeed (knots)	During ten seconds around time point
i	Angle difference between aircraft track and runway track (°)	At time point
f	Average flight path angle (°)	During ten seconds around time point
a	Average flight path acceleration (g)	During ten seconds around time point

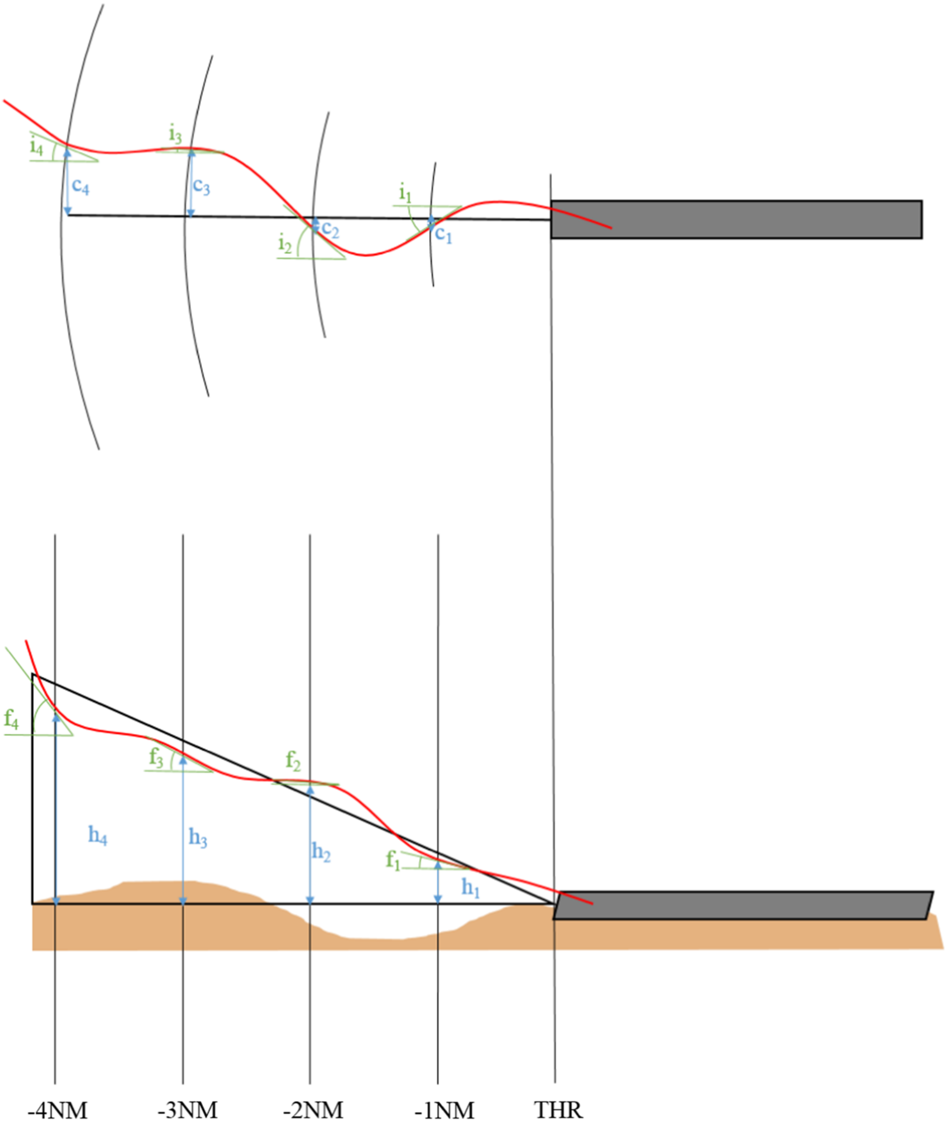

Fig. 3. Feature space illustration.
Source: Author.

(c_1, c_2, c_3, c_4 … c_{10}; h_1, h_2, h_3, h_4 … h_{10}; s_1, s_2, s_3, s_4 … s_{10}; i_1, i_2, i_3, i_4 … i_{10}; f_1, f_2, f_3, f_4 … f_{10}; a_1, a_2, a_3, a_4 … a_{10})

3.3.3. *Parameter volatility*

The volatility of the parameters described above affects the measurement at the time points. A highly volatile parameter (such as acceleration (*a*) in turbulent conditions) will have a reduced significance if it is only measured at one point in time because its value might be significantly different just one second later. For other parameters, such as height (*h*), which is normally monotonically decreasing and dampened by inertia, the effect of volatility is far less pronounced.

Therefore, for the highly volatile parameters speed (*s*), flight path angle (*f*) and acceleration along the flight path (*a*), the measurement is defined as the parameter rolling mean from five seconds before the relevant time point ($t - 5\,s$), to five seconds after ($t + 5\,s$).

3.4. *Creating and cleaning the dataset*

3.4.1. *Defining time points and measurements*

First, the time points were defined using the global parameter (i.e. a parameter available from all flights) “Ground Track Distance to Threshold (NM)” and defines the time point t_1 as the first location in the airborne interval when this distance is not ≥ 1.

Once all ten time points were defined, the six measurements for each time point, as defined in Table 1 above, could also be defined.

3.4.2. Processing the measurements

The operator's FDM software was then used to extract the measurements at each time point for 1.2 million flights. After successful processing, a random sample was manually checked for gross errors or missing data. Several adjustments to the precise definitions of the measurements were made in order to create data which precisely reflects the feature space as described above. After each adjustment, the flight data was reprocessed, taking several days to complete due to the volume of data.

The results were then exported as a single line per flight comma-separated values (CSV) of all measurements. Additionally, a unique flight id, flight month and year, fleet id, airport and runway were added for each flight. The resulting data file had a size of 1.12 GB.

3.4.3. Further processing of the dataset

For further processing, the dataset was loaded into RStudio, an integrated development environment (IDE) for R (RStudio Team, 2015). Both R and RStudio are open-source software and widely used among statisticians and data scientists. R allows users to create packages, which expand the abilities of the core R installation. Appendix B lists the versions of R and additional packages used.

The data.table package's function *fread* was used to read in the CSV file and convert it into an R data frame containing more than 60 million data points.

3.4.3.1. Flight data errors. Typical flight data errors are synchronization errors and cycling. A synchronization error occurs when some parts of the data are not recorded due to temporary recorder failures, electrical transients or other issues. In such cases, the flight data will show "jumps", since the data does not show a steady times series of recordings but rather connects the last value before the error with the first value after, leading to unrealistic immediate changes in altitudes, speeds etc. An example can be seen in Fig. 4.

Cycling occurs when certain parameters are not available, possibly during an entire flight or even several flights. Instead of recording "does not exist" (DNE) recorders will sometimes record a cycling pattern, in which the parameter is recorded as a quick succession of its minimum and maximum value. An example can be seen in Fig. 5, where the airspeed, ground speed and rotational speed of the low-pressure spool (N1) of two engines alternate between 0 and their respective maximum values of 512 knots, 1023 knots and 127.88%.

These two common errors, as well as others occurring randomly, can remain undetected during the regular post-processing and therefore lead to incorrect measurements. Since it is highly probable that such erroneous measurements will lead to high outlierness, the following data cleaning steps were performed.

3.4.3.2. Data cleaning. Data cleaning is generally a trade-off between thoroughness and minimizing the loss of valid data. When using strict filtering criteria, the filter will be able to remove most of the unwanted data, but at a high risk of eliminating valid data by mistake. A less strict filter, on the other hand, will preserve most of the valid data but also miss some of the erroneous values. Since this work is about finding outliers, which are characterized by unusual flight data values, the filter criteria were intentionally defined to contain wide margins. To avoid deleting valid data, the following limits were defined:

1. *c* (lateral distance from runway centreline): The data points are recorded based on the remaining ground track mileage until reaching the landing runway threshold. Therefore, the lateral offset from the runway centreline can never exceed the remaining distance until threshold. Consequently, each value bigger than the remaining mileage is removed.
2. *h* (height above runway threshold): An aircraft typically descends at an angle of 3° when approaching a landing runway, which equals an altitude loss of 318 feet per NM, as shown in Eq. (1).

$$\frac{6076\ \text{ft}}{1NM} * \tan 3 = 318\frac{ft}{NM} \tag{1}$$

It can be assumed that an aircraft will not be climbing towards a landing runway. Therefore, the lower limit at each mile is set equal to the airport elevation. Lower altitudes are rejected as data errors. Steeper approach angles are often observed due to various reasons, e.g. mismanaged energy situation during the descent or late descent clearances by Air Traffic Control (ATC). An assumption was made that no aircraft will be descending for 1 NM at more than five times the normal approach angle of 3°, defining a maximum altitude loss corresponding to this 15-degree limit at 1628 feet/NM as shown in Eq. (2).

$$\frac{6076\ \text{ft}}{1NM} * \tan 15 = 1628\frac{ft}{NM} \tag{2}$$

Consequently, the difference between each measured altitude was calculated and each flight which exceeded this limit was rejected.

3. *s* (average airspeed): The range of airspeeds depends on the aircraft type. The highest operating speed of all aircraft in this airline is the never exceed speed (VNE) of the Boeing 747 at 365 knots. A 30 knots margin is added and any airspeed above 395 knots is considered a data error. The lowest speed is the stall speed, which is normally increased by 30% to get the landing reference speed (VREF). The lowest VREF is 107 knots of an empty B737-500. Therefore, the lowest possible airspeed outside the stall is calculated 82 knots. Any speed below 82 knots is considered a flight data error.
4. *i* (angle difference between aircraft track and runway track): The angle between runway direction and the aircraft track can be of any value if the aircraft is still several NM from touchdown. For example, during a circling pattern, the aircraft might align with the landing runway at about 2 NM from touchdown, being at a 90°

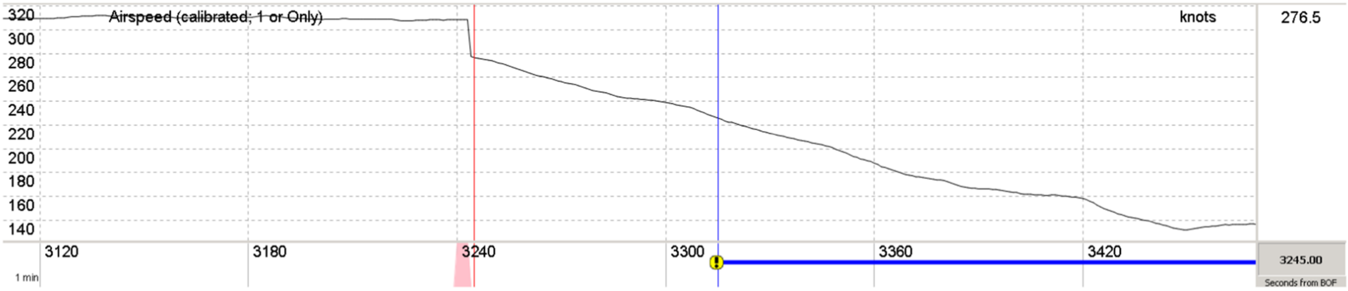

Fig. 4. Synchronization error: The red shaded area at the bottom indicates that software detected a synchronization problem, which caused a sudden sharp jump in the recorded airspeed. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)
Source: Author, taken from ADI EMS software.

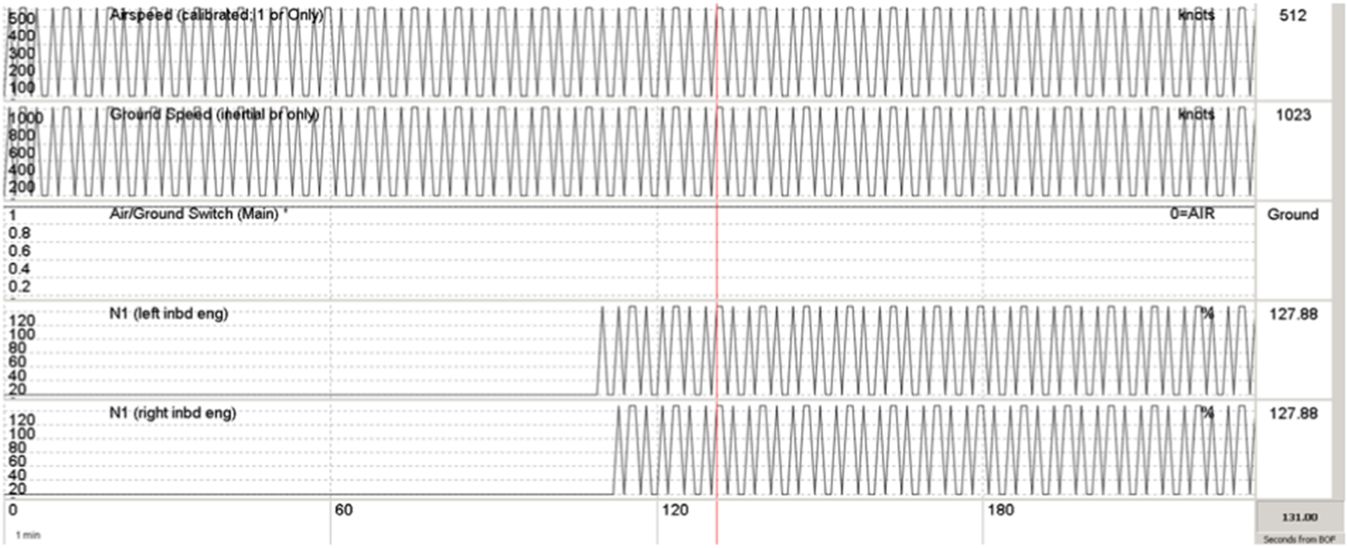

Fig. 5. Cycling of several recorded flight parameters.
Source: Author, taken from ADI EMS software.

intercept at 3 NM and possibly on downwind with a 180° difference in track at 4 NM. Therefore, the only value that can be cleaned is the angle at 1 NM. Here, it is assumed to be unrealistic when the angular difference is in excess of 90°.

5. f (average flight path angle): As previously stated the normal descent angle during approach is 3° and any value in excess of five times this angle is considered unrealistic. Hence, any value lower than $-15°$ should be removed. However, at the time of attempted removal and after the height values (h) had been cleaned, there were no such values in the dataset.
6. a (average flight path acceleration): During turbulence and when performing dynamic manoeuvres significant accelerations may occur. The dataset contained accelerations between -0.52 g and 0.84 g. As both these values were considered realistic no further cleaning was performed.

After performing these cleaning steps the number of flights was reduced by 225, resulting in a final number of flights of 1,097,943.

3.5. Performing the local outlier probability analysis

Before the LoOP analysis can be performed, the data has to be transformed into a suitable format as follows.

3.5.1. Standardization

The absolute values of the different measurements differ by orders of magnitude (e.g. airspeed and altitude). In order to avoid an influence of the magnitude of absolute values, the data was standardized. For the entirety of each measurement (e.g. all c_i values) the mean value and standard deviation were calculated. The mean was subtracted from each value, thereafter each value was divided by the standard deviation, see Eq. (3).

$$x_{i,standardized} = \frac{x_i - \bar{x}}{\sigma_x} \tag{3}$$

After standardization, the data has a mean of 0 and a standard deviation of 1. The outlier characteristics of the data are preserved by this method.

3.5.2. Weighting

If some measurements are more important than others they can be weighted in order to reflect these differences in importance. For example, data closer to the threshold could be weighted as being more important by multiplying each measurement with the inverse of the remaining ground track distance. Since the aim of this work is to uncover the hidden properties within the data, it was decided not to apply such a weighting as it would introduce a further element of human judgement.

3.5.3. Finding the number of clusters

The LoOP algorithm needs the number of clusters in the data as an input prior to starting its calculations.

The optimal number of clusters is ambiguous. It depends on the definition of clusters and several answers might be equally appropriate. However, each calculation requires the input of a single number of clusters. Running several calculations with different numbers of clusters was dismissed as being too inefficient. Instead, the number was determined by calculating the *within cluster sum of squared errors* (WSS) for 1 through 60 k-means clusters. With increasing numbers of clusters, less variance will appear within each cluster, but the difference between nearby clusters will also be reduced. The number of clusters was chosen for which a further increase of clusters did not significantly affect the reduction of WSS. This method was described by Everitt and Hothorn (2010) and the algorithm was adapted from Ben (2013).

Fig. 6 shows the WSS in the upper plot and the reduction of WSS when increasing the number of clusters by one in the lower plot. It can be seen that increasing the number of clusters to 17 does not decrease the WSS, so 16 clusters were chosen as input parameters for the LoOP algorithm.

3.5.4. LoOP analysis

The implementation of the LoOP algorithm used for this analysis is the "*Environment for Developing KDD-Applications Supported by Index-Structures*" (ELKI) java package developed at the Ludwig Maximilian University (LMU) in Munich (Schubert et al., 2015). Version 0.7.1 was obtained as executable java (JAR) archive from the LMU internet domain. ELKI is an open-source software package optimised for ML and knowledge discovery in databases (KDD). Its Local Outlier Factor implementation outperforms R implementations by factors of up to 280. This level of performance was desirable for conducting a LoOP analysis on the 60 million data points.

The most current Java Runtime Environment (JRE) version available at the time (JRE 1.8.0_51) was selected. The analysis was started using a submission script via the Altair Portable Batch System (PBS)

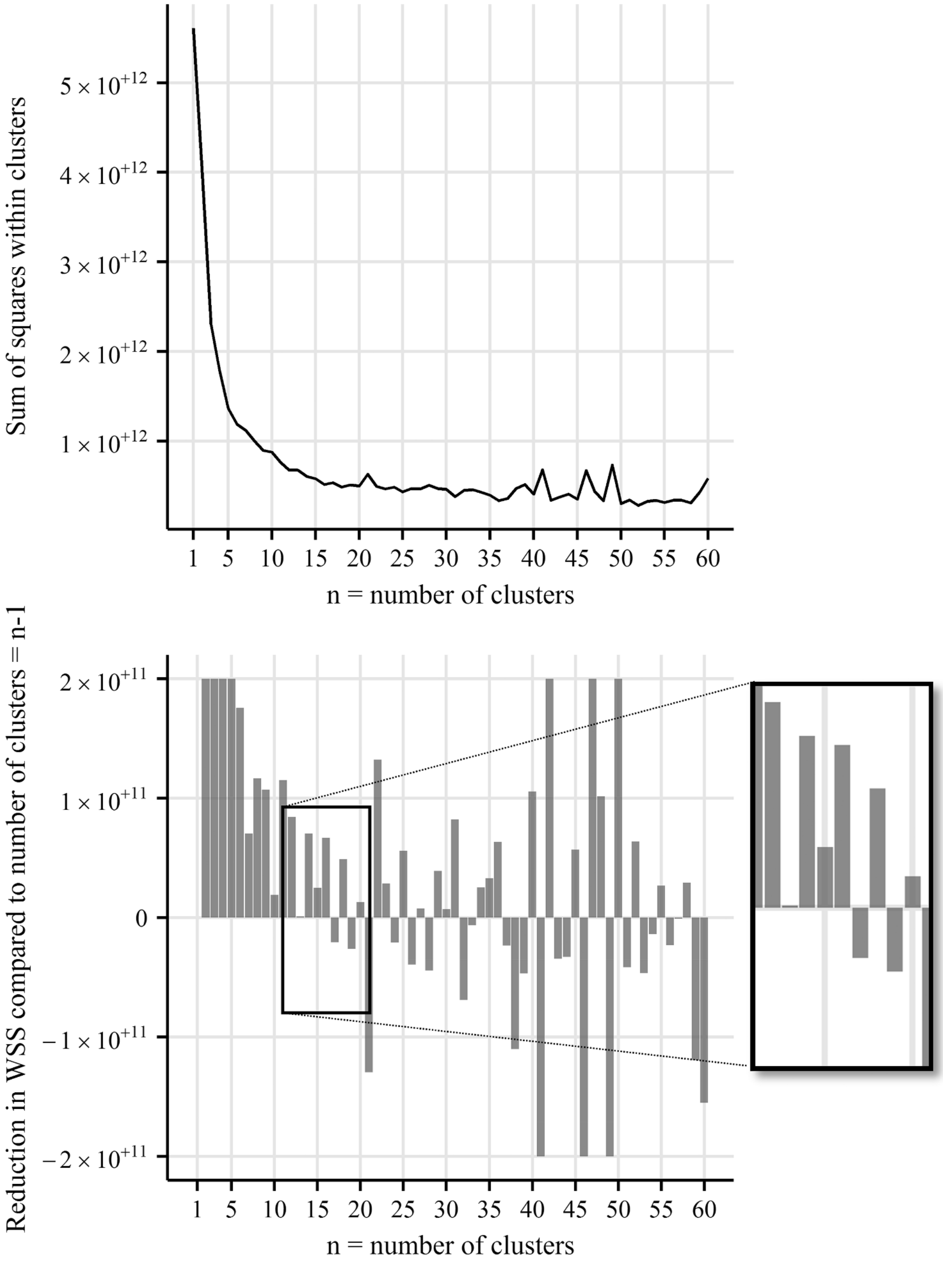

Fig. 6. Upper: Within groups sum of squared errors for 1–60 clusters. Lower: Change of within groups sum of squared errors compared to one cluster less; large values are truncated to illustrate small changes.
Source: Author.

Professional tool. This script allows the user to specify the number of Central Processing Units (CPU) for parallel computing. Requesting a higher number of CPUs reduces the available working memory per CPU. Since the LoOP algorithm has a high memory requirement, only one CPU was selected. There is a high chance for significant performance improvement when several CPUs are used.

The command used to run the analysis in ELKI was

```
java -jar elki-bundle-0.7.1.jar KDDCLIApplication
dbc.in “dataset.csv.gz”
algorithm outlier.lof.LoOP loop.kcomp 16
evaluator NoAutomaticEvaluation resulthandler ResultWriter
out FullLoOP
```

The calculation ran for 100 h 2 min and 54 s. It returned the Local Outlier Probability for each flight and these were read into R in order to evaluate the results.

3.6. Comparing the outlier with the standard exceedance detection events

The data of each flight in this airline is routinely filtered by a mature exceedance detection system (i.e. a typical FDM system). It compares defined parameters with pre-determined thresholds. If these thresholds are exceeded, an exceedance event is created. The software automatically labels the events with a severity on the scale of “*Information Only*”, “*Medium*” and “*High*”. It is the airline’s standard procedure to review only events of “*High*” severity, hence events of the severity “*Information Only*” and “*Medium*” are not considered in the following analysis.

At the time of this work, a total of 134 exceedance event types were available in the monitoring software, of which 80 were actually observed at the airline during the three years which were analysed. All exceedance events of the severity "*High*" are reviewed by a review pilot. The reviewer assesses the safety relevance and rates these events according to an adapted version of the Aviation Risk Management Solutions (ARMS) methodology (Aviation Risk Management Solutions, 2010).

The ARMS methodology rates the effectiveness of the remaining barriers between the observed event and the most credible accident scenario. The more effective the remaining barriers are, the less severe is the observed event.

The term "severity" is used for two different classifications during the review process in this airline. The automatically determined severity with the three levels (high, medium and information only) reflects the magnitude or the duration of the parameter exceedance. It is solely used to limit the number of exceedance events which have to be reviewed and focus on those which are most likely to be of high safety relevance. After the review, a severity assessment according to the ARMS methodology is conducted by the review pilot. This results in a severity rating based on human judgement on a five-level scale:

$$a.....f$$

where a represents highest severity and f lowest severity.

This severity rating is far more meaningful than the automatic three-level rating.

In order to compare the results of the LoOP methodology with the standard exceedance detection method, LoOP-based proxy-events called outlier events were created for all flights with a Local Outlier Probability of 0.99 or greater.

The choice of 0.99 was somewhat arbitrary, however the investigation of the top 1% seems sensible and it was expected to result in a manageable number of outliers for manual review.

Each proxy-event was reviewed and rated by review pilots of the airline according to the same methodology as the conventional exceedance detection events. There are ten review pilots, each spending between one third and one half of their duty time monitoring and analysing flight data, and the remainder flying. On average they have more than fifteen years of experience as active pilots in the operation and more than ten years of experience in FDM.

4. Results and discussion

For each flight, the Local Outlier Probability was obtained. All values are in the range from 0 to 1. In order to determine whether this value is meaningful to measure safety-relevant occurrences, the distribution of the score is analysed and compared to existing measurements of safety relevance.

4.1. Analysis of the LoOP scores

The histogram in Fig. 7 shows the distribution of the LoOP scores. The most common score was 0, indicating that the algorithm rated these flights as being embedded in the centre of clusters. This shows that the expected clustering happened and that the feature space created in this work did result in clusters containing the majority of the approaches. The mean value is approximately 0.3, and only 1% of the flights have a LoOP of greater than or equal to 0.8.

From this distribution, it seems reasonable to assume that flights which triggered FDM exceedance events would have higher LoOP scores i.e. flights with exceedances more likely to be outliers. To investigate this, Fig. 7 was reproduced using only those flights in which an exceedance had occurred. The resulting histogram is shown in Fig. 8.

It can be seen, that the distribution is shifted to the right towards higher scores. This shows that the LoOP created through the developed feature space is influenced by safety-relevant occurrences (exceedances) during the flight. The mean LoOP score is 0.56, higher than the 0.3 mean for all flights and even higher than the third quartile of the

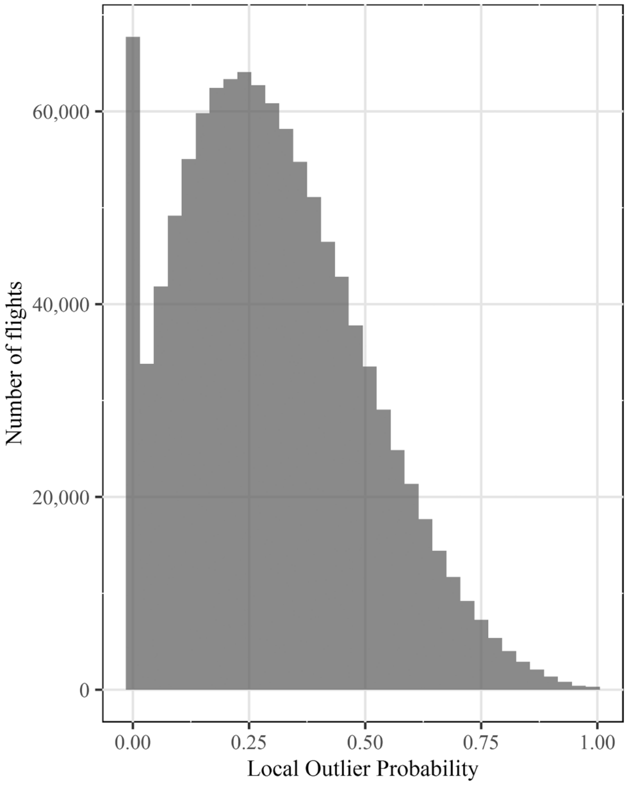

Fig. 7. Histogram of LoOP values.
Source: Author.

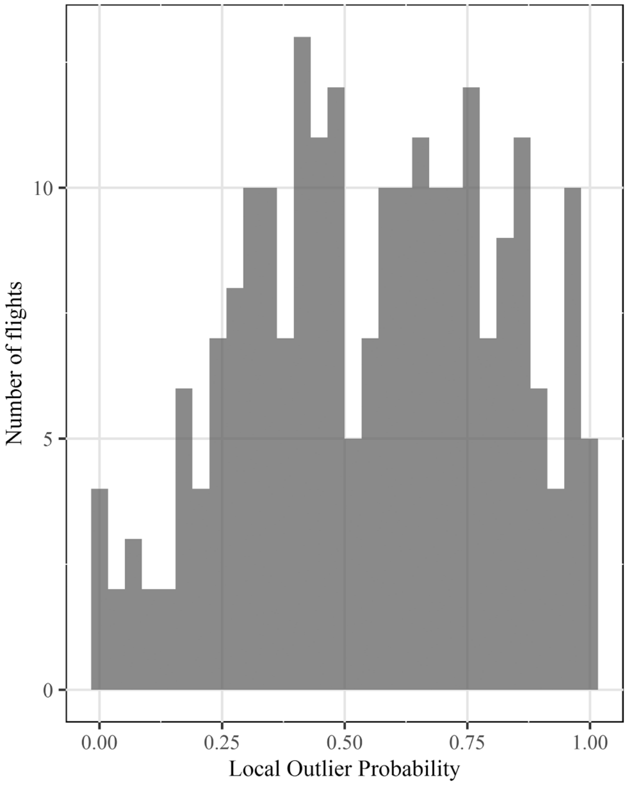

Fig. 8. LoOP score of reviewed exceedance events.
Source: Author.

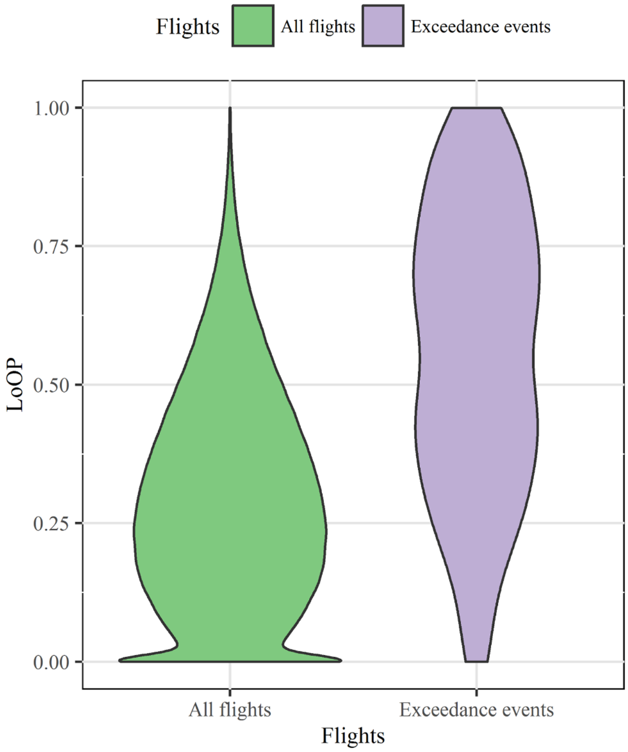

Fig. 9. LoOP scores of all flights vs. LoOP scores of flight with exceedance events.
Source: Author.

LoOP score distribution from all flights at 0.43.

The violin plot in Fig. 9 shows this difference in distributions. In this plot, the thickness of the coloured area is governed by a kernel density estimation, which represents a smoothened relative frequency of the LoOP score within the group. Many flights from the "all flights" group have a LoOP score of 0, which leads to a very wide lower end of the violin shape, and only very few achieve a score of 1, which explains the thin upper end. For the subset of flights with exceedance events this is almost reversed. Especially the wide upper end at the maximum LoOP score is noteworthy. While only very few flights from the complete population of all flights receive the maximum LoOP of 1.0, this score is not uncommon among the flights which already triggered an exceedance event.

Note that due to the relatively small number of flights with exceedances, a plot of non-exceedance flights would be completely indistinguishable from the "All flights" plot in Fig. 9.

This indicates a clear correlation between safety-relevant occurrences during a flight and the LoOP created by the feature space developed in this work. This is the first time that such a connection between safety-relevant occurrences and a ML score independent from classic exceedance detection has been shown for more than just a few selected example flights.

4.2. Comparison of outlier events and exceedance events

Of 22 flights found by the LoOP approach only five were known from the exceedance event method. This means that the new method found 17 new cases of safety-relevant occurrences in a database of flights which was already monitored by a mature FDM system. These novel findings are discussed in Section 4.4.

Table 2
Confusion matrix for event-based flight data analysis.

Classified as:	Safety-relevant	Uneventful
Truly safety-relevant	True positive	False negative
Truly uneventful	False positive	True negative

4.3. Performance evaluation

The task of finding safety-relevant flights can be considered as a classification task. The classifier is the event creation mechanism for either the exceedance event or the outlier event. The classification task is to distinguish between uneventful and safety-relevant flights.

The most common methods for performance evaluation of classifiers are based on the confusion matrix. A confusion matrix evaluates how many of the positive and negative classification results are correct (true) and incorrect (false). This is usually represented in a two-by-two table showing true positives, true negatives, false positives and false negatives. The confusion matrix for event detection is shown in Table 2.

From this basic division of results into four groups, the most common performance evaluations such as recall, precision, sensitivity, specificity or the F-score can be derived.

A basic requirement to create the confusion matrix and subsequently the measurements of performance described above is a "gold standard" classifier. This classification method is understood to uncover the "ground truth", which in this case is the column-wise distinction in Table 2, the differentiation between flights which actually had a safety-relevant occurrence and these which did not. The gold standard to define true positives and true negatives is the review pilot. Experienced reviewers are the best means available to distinguish between the two groups and can be assumed to reach the closest approximation to the ground truth. Review pilots are not perfect. There has not been any academic research into the inter-rater reliability of review pilots, nor is there an industry-wide common understanding of what constitutes a safety-relevant occurrence and what does not. However, in the airline supporting this research, many events are reviewed by at least two review pilots, a so-called "four-eyes principle", in order to improve the inter-rater reliability. The imperfection of this approach can be shown by one occurrence detected by both an outlier event and an exceedance event, which was rated with severity *b* from the exceedance event reviewer and severity *b-c* from the outlier event reviewer. Nevertheless, the distinction between true positives and false positives can be assumed to be sufficiently close to the ground truth.

However, it is not realistic to expect review pilots to distinguish between true and false *negatives*. This is an effect of the low prevalence of safety-relevant occurrences during airline operations. Similar difficulties have been described in network intrusion detection (Mane et al., n.d.) and epidemiology (Joseph et al., 1995). Out of the 1 million flights considered in this work only a few hundred created any event (outlier event or exceedance event) of severity *d* or higher. Even under a very pessimistic assumption that for every detected event there are 10 undetected events, the prevalence of safety-relevant occurrences is only in the order of magnitude of 0.001 or one in a thousand flights. In order to find a statistically significant number of false negatives, a review pilot would have to review tens or even hundreds of thousands of flights of which a vast majority would indeed by uneventful (true negatives) and find the sparsely distributed false negatives. This approach is not only cost prohibitive but would also be prohibitively error-prone.

Therefore, a comparison between the exceedance event and the outlier event methodology is only possible on the basis of true and false positives. The measurement "positive predictive value (PPV)" or "precision" uses the proportion of true positives among all positives as a performance measure. It is calculated as shown in Eq. (4).

$$PPV = \frac{number\ \ of\ \ true\ \ positives}{number\ \ of\ \ true\ \ positives + number\ \ of\ \ false\ \ positives} \tag{4}$$

The PPV of the exceedance event system is 0.23. This value is based on 1982 exceedance events with severities of d and above, and 6654 false positives, which consist of 483 exceedance events of lower severities and 6171 exceedance events with data errors, undesired event triggering or other technical problems.

Among the 134 outlier events that were created, there were 112 events which could be classified as false positives. This results in a PPV of 0.16, which is significantly lower than the PPV of the exceedance event approach. This is not surprising since the exceedance event definitions and algorithms in this airline have evolved over the last 15 years and were constantly improved in order to minimise the proportion of false positives. Nevertheless, the number of false positives resulting from the outlier event method is high and therefore the reasons are investigated.

4.3.1. False positives in outlier events

The false positives can be divided into two subgroups: data errors and less relevant outliers. Even though data cleaning methods were used in order to limit the amount of data errors among the outlier events these methods could not eliminate all instances of erroneous data. In 27 cases a specific data error which was not expected during data cleaning occurred: The recorded aircraft track temporarily indicated 0 instead of the actual value. This caused the calculated parameter i (Angle Difference between Aircraft Track and Runway Track) to become equal to the runway orientation, which can take any value between 0 and 360°, while a realistic runway for the close-in time points would normally not exceed ten degrees. Fig. 10 illustrates how this kind of data error causes an incorrect angular difference measurement of approximately 280°.

Without this group of false positives, which could be filtered out by a rate filter during data generation, the PPV would be 0.21, thus similar to the exceedance events (0.23). The PPV for exceedance events is acceptable to the supporting airline and viewed as practicable, which suggests that the PPV achieved through the methods described here would also be acceptable. It is worth noting that the methods described in this paper are intended to complement, rather than replace, exceedance events.

The second subgroup of false positives among the outlier events are flights which are indeed outliers according to the chosen detection approach, which means that there are no data errors or other technical factors causing a too high LoOP, but which are not safety-relevant.

This is a fundamental disadvantage of the outlier approach when it is used solely for flight safety purposes. While it is very likely that a safety-relevant flight will be an outlier, the opposite is not necessarily true. A flight's trajectory can be very unusual, for example due to unusual but not hazardous weather conditions, ATC requirements etc., but not unsafe at all. Finding these flights among the ones with high LoOPs is the systemic drawback of being able to find unexpected and novel safety-relevant occurrences. However, even when outliers are irrelevant for safety purposes, they might still be of interest to other stakeholders in an airline, such as fuel efficiency or training departments. Both the advantage and disadvantage, result from the fact that thresholds for triggering a LoOP are created without any expert flight safety knowledge, but solely on the basis of comparison with a large number of other flights.

4.4. Safety knowledge gain beyond single occurrence detection

The comparison of PPVs alone is of limited value. A high PPV avoids unnecessary review effort by the review pilots, but it has no direct positive influence on flight safety. The reduction of false negatives, on the other hand, can lead to novel insights into the risks of flight operations and thereby contribute to an increase in safety. The main benefit of using an outlier event approach is not to outperform the exceedance event system in PPV or other performance measures, but to uncover new threats which were not previously considered and hence for which no exceedance events were developed.

Out of the 22 occurrences detected by outlier events with a severity of *d* or higher only five were previously known from exceedance events. For the 17 newly detected occurrences it is of great interest, whether they can yield truly new insights or whether they are just slightly different variants of known occurrence categories which could have also been discovered by marginally adjusted exceedance event definitions.

4.4.1. Groupings in the high severity outlier events

When reviewing the outlier events two groups of unusual manoeuvres were identified, for which no specific exceedance events exist.

The first group is visualised in Fig. 11. It consists of three approaches during which a 360° turn on final was performed, one of them at very low altitude.

Because a 360° turn is not inherently unsafe the flights were ranked with rather low event severities of *c* for the lowest 360° turn, *d-e* for one at a greater height and *e* for the highest.

Consequently, two out of three of these flights can be considered false positives when focusing on safety-relevant occurrences and only the lowest one is among the 22 outlier events of safety interest. However, all three show a similar manoeuvre which may or may not be safety-relevant but for which no exceedance event exists. And as the manoeuvre at the lowest height shows the occurrence can be of safety interest. The lack of a specific exceedance event category for this manoeuvre is because neither the airline's FDM department or FDM software supplier had considered such an event before. Understandably so, given the almost infinite potential safety deviations from normality, but this highlights the need for methods such as ML which do not rely on predefined specific event searches.

Similarly, there are three flights which show an avoidance manoeuvre on the final approach track. The deviations from the expected straight-in approach are illustrated in Fig. 12. These approaches were assigned with event severities between *c-d* and *e*, again illustrating that such manoeuvres might be of interest, but no adequate exceedance event exists to capture these flights.

This shows that the concept of using outlierness scores is capable of

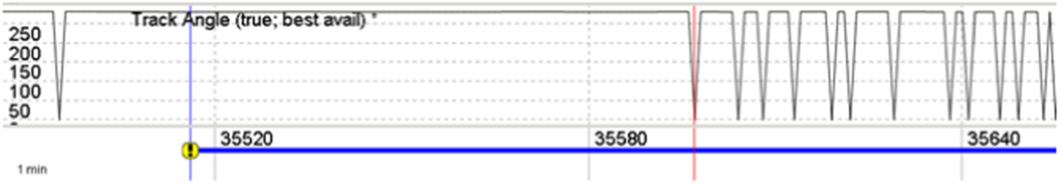

Fig. 10. Measured Track Angle erroneously indicates 0 for short periods of time.
Source: Author, taken from ADI EMS software.

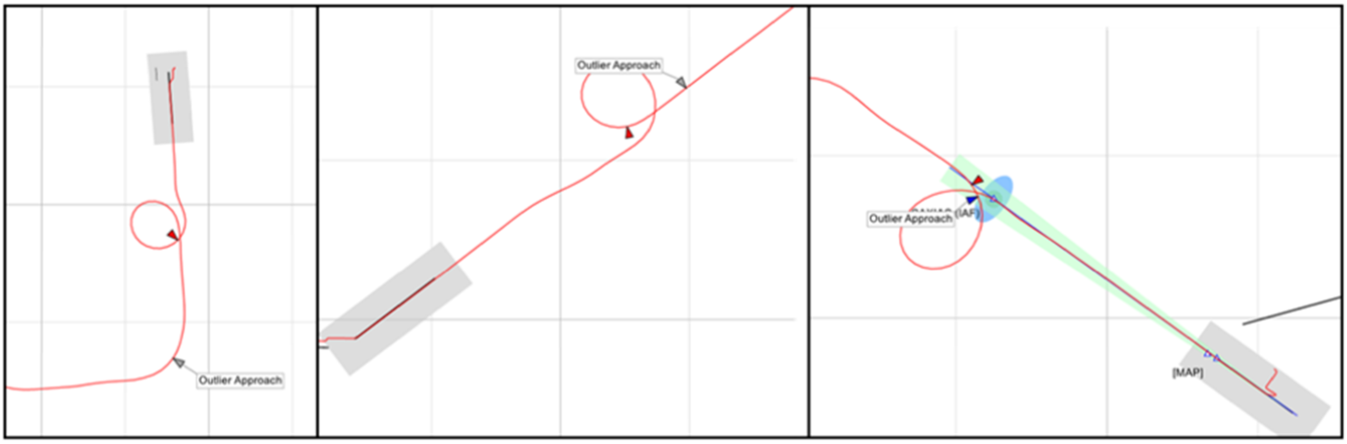

Fig. 11. Approach trajectory schematic representation from the ADI EMS software of the three 360° turns on final.
Source: Author.

discovering novel and relevant occurrence categories in a mature FDM system, thereby reducing the number of undetected safety-relevant occurrences. It is, and will remain, impossible to create specific exceedance event types for every relevant occurrence which may occur at some point in the future. If, however this novel occurrence is significantly different from all or almost all other flights, an outlier detection algorithm can find it in the flight data.

4.4.2. *Outlierness as an aggregate measurement of risk across occurrence categories*

There are two examples of flights which performed unstable approaches which were not detected by the exceedance event system because each single factor contributing to the instability of the approach was not enough to trigger the "high" criterion on the automatic severity classification.

The first is an Airbus A320, performing an approach to a major European hub airport, which triggered the medium-severity exceedance events "*Late Flap Extension*" and "*Late Gear Extension*".

The second is also an Airbus A320 flying into a Western European coastal airport, which triggered five medium-severity exceedance events:

> *"Late Flap Extension", "Late Gear Extension", "GPWS: Glideslope", "Below Desired Glide Path on Approach" and "Unstable Approach"*

For none of the above-mentioned exceedance events was the exceedance sufficient to trigger a "high severity event" which would have prompted a review of the flight. However, the LoOP from the sixty-dimensional feature space consider the overall outlierness of the approaches and resulted in values of 0.9945 and 0.9940. When the outlier events were created and the flights reviewed, both received a severity rating of *b-c*, which is the second highest category observed for any approach and would have triggered an investigation.

This shows a fundamental advantage of the outlier event concept: The flight is described and ranked as the entirety of all the measurement in the feature space. It is not necessary that a single limit is exceeded by a wide margin in order to trigger the specific event for a single occurrence category. Instead, what would be called risk- or severity-aggregation in classical exceedance event systems is inherently a part of the outlierness concept. Each deviation contributes to one overall score, even if the measurements in the feature space describe fundamentally different properties of the flight.

4.5. *Development*

This work has shown that ML methods can be successfully applied to large flight datasets to discover new safety knowledge. An obvious, albeit self-imposed, limitation is that only the period of flight from 4NM to touchdown to touchdown is used. It would be interesting to see if the same feature set could identify unusual occurrences in other flight phases e.g. departures, say take-off to 4NM during the initial climb.

There is no reason why the feature space could not be adapted and tailored to other phases of flight. For example, it might be interesting to adapt the feature space and examine the descent phase for arrivals at

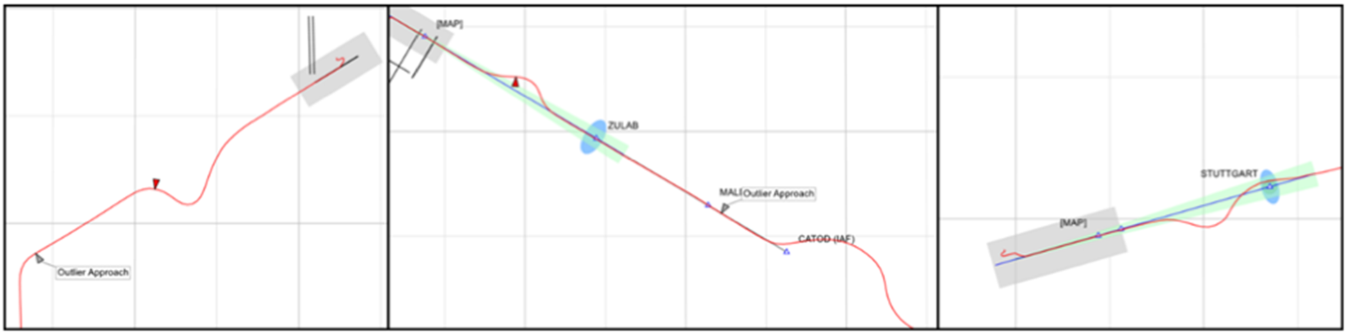

Fig. 12. Approach trajectory schematic representation from the ADI EMS software of the three avoidance manoeuvres on final.
Source: Author.

specific airports, or specific departure/arrival airport pairings. This may yield more in the way of operational efficiencies outliers, such as arrivals with particularly high fuel usage.

This work made use of a very large dataset of over 1 million flights, showing that the methods are practicable in a large airline dealing with a large number of flights. However, it would also be interesting to see if the methods could be used effectively for smaller airlines with fewer flights. It is the authors' intention to do this in a future paper and at the same time make effectiveness comparisons between different feature spaces.

In more general, the large amounts of data generated through FDM programmes would seem to make it a good candidate for exploring other ML methods, such as neural networks. This is particularly so, because this domain is not as critical as say, medical diagnosis, the mechanism for how new safety knowledge is discovered is of fairly limited importance. The methods act as more of a "flag" that there is something unusual that an expert should have a look at, as is the case now with exceedance based FDM. As long as whatever method is used does not result in high numbers (a higher rate than existing methods) of false positives, it should be practicable.

This research made use of a high-performance computing environment, however only a single CPU was used which resulted in a runtime of 100 h. It is expected that the runtime could be significantly reduced by using multiple CPUs, and the authors anticipate that it would be feasible to run the ML on less specialised hardware in a production environment i.e. the safety department of an airline.

5. Conclusions

The origins of flight data monitoring can be traced back to the 1970's with the adoption of quick access flight data recorders on Hawker Siddeley Trident aircraft. The primary analysis methodologies have changed little since then, with basic exceedance detection and measurements analysis being the common methods used today. These current methods rely on aircraft operators to predefine algorithms to detect anomalies. The range of these algorithms has grown over the decades in response to incidents that were not detected by existing algorithms at the time. However it is impossible to foresee, and therefore create new algorithms, for every potential future incident, and this limits the capability of FDM to help provide new safety knowledge.

This research has shown that the analysis of existing FDM data can be enhanced through the application of the LoOP algorithm. The comparison with the traditional FDM showed that this outlier approach can find flights which were missed by the traditional system, detecting approach abnormalities for which no exceedance event existed.

This research focussed on the approach phase of flight, however the methods described here could be extended to other phases of flight by defining new feature spaces. In addition, different time points and measurements could be defined. It is envisaged that the methods could be used to help highlight anomalies related to the economical operation of aircraft, in addition to safety anomalies described here.

Furthermore, this method is useful whenever a large number of possibly very different flights has to be monitored by a superordinate institution, e.g. a national aviation authority or a risk management department of a group of airlines.

This research achieved its aim of identifying safety-related occurrences that are currently undetected by typical FDM systems and shows that the methods used can be used to complement existing technologies. However, it is recommended that future work compares the performance of other outlier detection approaches.

Appendix A. Abbreviations used

ABOD	Angle-Based Outlier Detection
APM	Automated Parameter Measurement
ARMS	Aviation Risk Management Solutions
ATC	Air Traffic Control
CFIT	Controlled flight into terrain
CPU	Central Processing Unit
CSV	Comma Separated Values
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DNE	Does Not Exist
ELKI	Environment for Developing KDD-Applications Supported by Index Structures
EMS	Event Measurement System
FDM	Flight Data Monitoring
GB	Gigabyte
GUI	Graphical User Interface
HPC	High Performance Computing
IDE	Integrated Development Environment
JAR	Java Archive
JRE	Java Runtime Environment
KDD	Knowledge Discovery in Databases
kNN	k-Nearest Neighbours
LMU	Ludwig Maximilian University
LOC-I	Loss Of Control Inflight
LOF	Local Outlier Factor
LoOP	Local Outlier Probability
ML	Machine Learning
N1	Rotational speed of engine's low-pressure spool (percent of nominal value)
PBS	Portable Batch System
PPV	Positive Predictive Value
RE	Runway Excursions
SQL	Structured Query Language
SVM	Support Vector Machine
VNE	Never Exceed Speed
VREF	Reference Landing Speed
WSS	Within Cluster Sum of Squares

Appendix B. Table of R packages used

See Table B1.

Table B1
Versions of R and packages used.

Name	Version	Purpose	Author
R (core)	3.2.3	Basic R application	R Core Team (2017)
NISTunits	1.0.0	Unit conversion	Gama (2014)
dplyr	0.5.0	Restructuring data	Wickham and Francois (2016)
tidyr	0.5.1	Restructuring data	Wickham (2016)
data.table	1.9.6	Reading in large amounts of data	Dowle et al. (2015)
ggplot2	2.1.0	Creating graphs	Wickham (2009)
RColorBrewer	1.1–2	Colouring graphs	Neuwirth (2014)
cowplot	0.6.2	Arranging graphs	Wilke (2016)
scales	0.4.0	Adjusting graph axis	Wickham (2016b)
extrafont	0.17	Adjusting fonts in graphs	Winston Chang (2014)

Appendix C. Orthogonal distance of the aircraft from the extended runway centreline (*c*)

This distance is calculated at each time point *t* by first determining the bearing from the runway threshold (*R*) to the aircraft (*A*), based on the runway threshold coordinates and on the recorded aircraft position. The calculation is as follows:

$$\beta = a\tan 2(X, Y) \tag{C1}$$

where β is the bearing from point *R* to point *A*, and *X* and *Y* are two quantiles as follows:

$$X = \cos\theta r.\ sin\Delta L$$

$$Y = \cos\theta a \cdot \sin\theta r - \sin\theta a \cdot \cos\theta r \cdot \cos\Delta L$$

where θa is the latitude of the aircraft θr is the latitude of the runway threshold and ΔL is the difference between the longitudes of points *R* and *A* (i.e. *lon R* – *lon A*). (Note that latitudes and longitudes are in radians.)

The distance c_n at time point t_n is given by:

$$c_n = d_n \sin\alpha_n \tag{C2}$$

where

d_n is the distance to the runway threshold at time points t_n

α_n is the angle between the aircraft's bearing to the runway threshold and the runway heading, given by

$$\alpha_n = |\beta - (runway\ \ heading - \pi)|\ mod\ \ 2\pi \tag{C3}$$

Runway headings and coordinates can be found in national Aeronautical Information Service Publications.

References

Airbus, 2016. A statistical analysis of commercial aviation accidents 1958–2016.

American Airlines, 2016. Snapshot of American in Dallas/Fort Worth.

Aviation Risk Management Solutions, 2010. The ARMS methodology for operational risk assessment in aviation organisations.

Ben, 2013. Cluster analysis in R: determine the optimal number of clusters [WWW Document]. URL < https://stackoverflow.com/questions/15376075/cluster-analysis-in-r-determine-the-optimal-number-of-clusters > (accessed 12.12.17).

Biswas, G., Mack, D., Mylaraswamy, D., Bharadwaj, R., 2013. Data mining for anomaly detection NASA STI program … in profile 2013–217973.

Boeing, 2016. Statistical summary of commercial jet airplane accidents, Boeing commercial airplanes.

Breunig, M.M., Kriegel, H.-P., Ng, R.T., Sander, J., 2000. LOF: identifying density-based local outliers. In: Proc. 2000 Acm Sigmod Int. Conf. Manag. Data, pp. 1–12. doi: 10.1145/335191.335388.

Campos, G.O., Zimek, A., Sander, J., Campello, R.J.G.B., Micenková, B., Schubert, E., Assent, I., Houle, M.E., Fuernkranz, J.B., 2016. On the evaluation of unsupervised outlier detection: measures, datasets, and an empirical study. Data Min. Knowl. Disc. 30, 891–927. https://doi.org/10.1007/s10618-015-0444-8.

Civil Aviation Authority, 2013. CAP739 - flight data monitoring.

Das, S., Li, L., Srivastava, A., Hansman, R.J., 2012. Comparison of algorithms for anomaly detection in flight recorder data of airline operations. In: 12th AIAA Aviat. Technol. Integr. Oper. Conf. 14th AIAA/ISSMO Multidiscip. Anal. Optim. Conf. doi: 10.2514/6.2012-5593.

Dowle, M., Srinivasan, A., Short, T., with contributions from R Saporta, S.L., Antonyan, E., 2015. data.table: Extension of Data.frame.

EASA, n.d. Acceptable Means of Compliance (AMC) and Guidance Material (GM) | EASA [WWW Document]. URL < https://www.easa.europa.eu/document-library/acceptable-means-of-compliance-and-guidance-materials/group/part-oro—organisation-requirements-for-air-operations#group-table > (accessed 12.12.17).

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.

Everitt, B., Hothorn, T., 2010. A Handbook of Statistical Analyses Using R. CRC Press.

Federal Aviation Administration, 2004. AC 120-82 - flight operational quality assurance.

Gama, J., 2014. NISTunits: fundamental physical constants and unit conversions from NIST.

Goldstein, M., 2012. FastLOF: an expectation-maximization based local outlier detection algorithm. In: Proceedings of the 21st International Conference on Pattern Recognition. International Conference on Pattern Recognition (ICPR-2012), 21st, November 11–15, Tsukuba, Japan, pp. 2282–2285.

Hodge, V.J., Austin, J.I.M., 2004. A survey of outlier detection methodologies. Artif. Intell. Rev. 22, 85–126. https://doi.org/10.1007/s10462-004-4304-y.

Hollnagel, P.E., 2014. Safety-I and Safety–II: The Past and Future of Safety Management. Ashgate.

Jesse, C., 2011. Intelligent Analysis of Aircraft Flight Data Parameters. University of Portsmouth.

Joseph, L., Gyorkos, T.W., Coupal, L., 1995. Bayesian estimation of disease prevalence and the parameters of diagnostic tests in the absence of a gold standard. Am. J. Epidemiol. 141, 263–272. https://doi.org/10.1093/oxfordjournals.aje.a117428.

Kononenko, I., 2001. Machine learning for medical diagnosis: history, state of the art and perspective. Artif. Intell. Med. 23, 89–109. https://doi.org/10.1016/S0933-3657(01)00077-X.

Kotsiantis, S.B., Zaharakis, I.D., Pintelas, P.E., 2006. Machine learning: a review of classification and combining techniques. Artif. Intell. Rev. 26, 159–190. https://doi.org/10.1007/s10462-007-9052-3.

Kriegel, H.-P., Kröger, P., Schubert, E., Zimek, A., 2009. LoOP. In: Proceeding of the 18th ACM Conference on Information and Knowledge Management – CIKM '09, CIKM '09. ACM, New York, NY, USA, pp. 1649. https://doi.org/10.1145/1645953.1646195.

Kriegel, H.-P., Shubert, M., Zimek, A., 2008. Angle-based outlier detection in high-dimensional data. In: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining – KDD 08, KDD '08. ACM, New York, NY, USA, pp. 444. https://doi.org/10.1145/1401890.1401946.

Laurikkala, J., Juhola, M., Kentala, E., Lavrac, N., Miksch, S., Kavsek, B., 2000. Informal identification of outliers in medical data. In: Fifth Int. Work. Intell. Data Anal. Med. Pharmacol. pp. 20–24.

Lazarevic, A., Ertoz, L., Kumar, V., Ozgur, A., Srivastava, J., 2003. A comparative study of anomaly detection schemes in network intrusion detection. In: Proc. 2003 SIAM Int. Conf. Data Min. pp. 25–36. doi: 10.1137/1.9781611972733.3.

Li, Lishuai, Hansman, R.J., 2013. Anomaly detection in airline routine operations using flight data recorder data.

Li, L., Das, S., John Hansman, R., Palacios, R., Srivastava, A.N., 2015. Analysis of flight data using clustering techniques for detecting abnormal operations. J. Aerosp. Inf. Syst. 12, 587–598. https://doi.org/10.2514/1.I010329.

Li, L., Hansman, R.J., Palacios, R., Welsch, R., 2016. Anomaly detection via a Gaussian Mixture Model for flight operation and safety monitoring. Transp. Res. Part C Emerg. Technol. 64, 45–57. https://doi.org/10.1016/j.trc.2016.01.007.

Mane, S., Srivastava, J., Hwang, S.-Y., Vayghan, J., n.d. Estimation of false negatives in classification*.

Markou, M., Singh, S., 2003a. Novelty detection: a review—part 1: statistical approaches. Signal Process. 83, 2481–2497. https://doi.org/10.1016/j.sigpro.2003.07.018.

Markou, M., Singh, S., 2003b. Novelty detection: a review—part 2: neural network based approaches. Signal Process. 83, 2499–2521. https://doi.org/10.1016/j.sigpro.2003.07.019.

Marr, B., 2015. Big Data: Using Smart Big Data, Analytics and Metrics to Make Better Decisions and Improve Performance. Wiley.

Matthews, B., Das, S., Bhaduri, K., Das, K., Martin, R., Oza, N., 2013. Discovering anomalous aviation safety events using scalable data mining algorithms. J. Aerosp. Inf. Syst. 10, 467–475. https://doi.org/10.2514/1.I010080.

Mendes, H., 2012. Study of mathematical algorithms to identify abnormal patterns in aircraft flight data.

Mitchell, T., 1997. Machine Learning (McGraw-Hill International Editions Computer Science Series). McGraw-Hill.

Nanduri, A., Sherry, L., 2016. Anomaly detection in aircraft data using Recurrent Neural Networks (RNN). In: 2016 Integr. Commun. Navig. Surveill. 5C2-1-5C2-8. doi: 10.1109/ICNSURV.2016.7486356.

Neuwirth, E., 2014. RColorBrewer: ColorBrewer Palettes.

Pimentel, M.A.F.F., Clifton, D.A., Clifton, L., Tarassenko, L., 2014. A review of novelty detection. Signal Process. 99, 215–249. https://doi.org/10.1016/j.sigpro.2013.12.026.

R Core Team, 2017. R: A language and environment for statistical computing. R Found. Stat. Comput. doi: ISBN 3-900051-07-0.

RStudio Team (2015). RStudio: Integrated Development for R. RStudio Inc, Boston, MA. URL < http://www.rstudio.com/ > .

Schubert, E., Koos, A., Emrich, T., Ufle, A., Schmid, K.A., Zimek, A., 2015. A framework for clustering uncertain data. In: Proceedings of the VLDB Endowment, vol. 8, no. 12.

Smart, E., 2011. Detecting abnormalities in aircraft flight data and ranking their impact on the flight.

Tax, D.M.J., Ypma, A., Duin, R.P.W., 1999. Pump failure detection using support vector data descriptions. In: Hand, D.J., Kok, J.N., Berthold, M.R. (Eds.), Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). Springer, Berlin, Heidelberg, pp. 415–425. https://doi.org/10.1007/3-540-48412-4_35.

United Airlines, 2016. United - Newsroom - Corporate Fact Sheet [WWW Document]. URL < http://newsroom.united.com/corporate-fact-sheet > (accessed 12.12.17).

West, M.M., McCluskey, T.L., 2000. The application of a machine learning tool to the validation of an air traffic control domain theory. In: Proceedings 12th IEEE Internationals Conference on Tools with Artificial Intelligence. ICTAI 2000. IEEE Comput. Soc, pp. 414–421. https://doi.org/10.1109/TAI.2000.889902.

Wickham, H., 2016a. tidyr: easily tidy data with 'spread( )' and gather( )' functions.

Wickham, H., 2016b. scales: scale functions for visualization.

Wickham, H., 2009. ggplot2: Elegant Graphics for Data Analysis. Springer, New York.

Wickham, H., Francois, R., 2016. dplyr: A Grammar of Data Manipulation.

Wilke, C.O., 2016. cowplot: Streamlined Plot Theme and Plot Annotations for "ggplot2.".

Winston Chang, 2014. extrafont: Tools for using fonts.

Zhao, K., Xie, Y., Tsui, K.L., Wei, Q., Huang, W., Jiang, W., Li, Y., Cho, S., Kim, S.B., Liu, K., Shi, J., Jeong, Y.S., Kim, B., Tong, S.H., Chang, I.K., Jeong, M.K., Charruaud, F., Li, L., 2015. System informatics: from methodology to applications. IEEE Intell. Syst. 30, 12–29. https://doi.org/10.1109/MIS.2015.111.