

CRANFIELD UNIVERSITY

VIVIAN ROWOLI IGENEWARI

**FRAMEWORK FOR ANOMALY DETECTION OF FLIGHT-CREW DEVIATION
FROM STANDARD OPERATING PROCEDURES: A DATA ANALYTICS
APPROACH**

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING

Integrated Vehicle Health Management (IVHM)

DOCTOR OF PHILOSOPHY

Academic Year: 2017- 2022

Supervisor: Dr. Steve King
Associate Supervisor: Prof. Ian K. Jennions
May 2022

CRANFIELD UNIVERSITY

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING

Integrated Vehicle Health Management (IVHM)

DOCTOR OF PHILOSOPHY

Academic Year: 2017- 2022

VIVIAN ROWOLI IGENEWARI

**FRAMEWORK FOR ANOMALY DETECTION OF FLIGHT-CREW DEVIATION
FROM STANDARD OPERATING PROCEDURES: A DATA ANALYTICS
APPROACH**

Supervisor: Dr. Steve King
Associate Supervisor: Prof. Ian K. Jennions
May 2022

This thesis is submitted in partial fulfilment of the requirements for the degree of
PhD

© Cranfield University 2022. All rights reserved. No part of this publication may be reproduced
without the written permission of the copyright owner.

ABSTRACT

Deviations from Standard Operating Procedure form a significant part of aviation incidents today involving loss of lives and other related costs. Previous work tailored towards detecting procedure deviations in flight operations have primarily been rule-based. The current method being used by airlines to detect operational, component fault and crew action anomalies within flight data is a rule-based Exceedance Detection technique which is only able to flag up known flight abnormalities. Lately, Anomaly Detection methods have been introduced to find, not just known, but unknown anomalies that deviate from the expected normal flight profile. There is a need to explore flight data using anomaly detection methods to detect subtle underlying misunderstandings of the flight crew in relation to deviations from laid down procedures which do not lead to incidents, under most conditions, or are hard to detect by the state-of-the-art method. However, these detection methods are limited in the type of anomalies they can find when implemented individually on heterogeneous flight dataset thereby missing critical anomalous flight incidents. In this work, Flight Data Recorder data of a fleet from a United Kingdom airline and a structurally similar publicly available dataset from the National Aeronautics & Space Administration are used. This study proposes a framework integrating an Ensemble anomaly detection technique (combining individual anomaly detection techniques into a single method) and a Case Based Reasoning system. The findings reveal that combining existing anomaly detection methods into an Ensemble can detect a wider variety of anomalies that were not flagged by individual methods. Also, the proposed reasoning design aims to filter for procedure deviations from the pool of anomalous incidents detected by the Ensemble. Detecting these procedure deviations is not just aimed at complementing crew training, improving procedures, and understanding automation design to put in place mitigation strategies but also to aid accident investigations by informing of accident flights with procedure deviations that may have been contributing factors.

Keywords:

Flight data, Standard Operating Procedures, Anomaly Detection, Ensemble methods, Machine Learning, Data Analytics, Case-Based Reasoning

I am dedicating this work to my loving and caring husband, Princewill and to our three lovely children Pamnigey, Pamenaan and Pamnieela. My P's ensure my life remains eventful through and through.

ACKNOWLEDGEMENTS

I am grateful to my Creator, the Jehovah El Roi, in whom I live, I move and have my being.

I am indeed very thankful to my supervisor, Dr. Steve King. I am a more confident Data Scientist and researcher because of his expert guidance and tutoring. ‘Doc’ has been a solid pillar throughout this journey spurring me on every step of the way with his constant positive outlook and encouragement at my lowest points.

I would also like to thank Prof. Ian Jennions for his immensely useful feedback that has often challenged me to think ‘out of the box’. I also owe ‘Prof’ a world of gratitude for his patience and support especially at the initial stages of this journey.

Dr. Zakwan Skaf was very instrumental in starting me out on this journey and for this I am very thankful.

My appreciation also goes to my sponsor in Nigeria, The Petroleum Technology Development Fund (PTDF) for funding this project and extending the support when I needed it. My colleagues from the IVHM Centre, specifically, Cordelia, Nithya, Shaf, Alexslis and David simply made the entire journey worthwhile with their cheerfulness and being my cheerleaders. My support network is huge, but I especially would like to extend my gratitude to the Eduns, Nwafor, Alozie and to my dear Molly.

My family played a big part in my research spurring me on with words and prayers. I especially, thank my number one fan, my husband, Princewill. He has been an awesome team player, juggling many hats when I could not take my eyes off the computer screen or get out of bed. At times when I am brain-blocked, his suggestions have been highly beneficial to keep me progressing. Through the COVID-19 pandemic, I was able to learn once again to value everyone that has been positioned in my life and I am very thankful for them all.

TABLE OF CONTENTS

ABSTRACT ii

LIST OF FIGURES xv

LIST OF TABLESxix

LIST OF ABBREVIATIONSxxi

1 Introduction 26

 1.1 Research Motivation 28

 1.1.1 Flight Data 28

 1.1.2 Anomaly Detection 29

 1.1.3 Standard Operating Procedures 30

 1.2 Aim and Objectives 31

 1.3 Methodology..... 32

 1.3.1 Phase 1: Literature Review 32

 1.3.2 Phase 2: Data Analytics and Labelling 34

1.3.3 Phase 3: Ensemble Anomaly Detection	35
1.3.4 Phase 4: Standard Operating Procedure Design.....	36
1.3.5 Phase 5: FAD-SOP Evaluation and Conclusion	37
1.4 Thesis Structure	37
1.4.1 Chapter 2: A survey of Flight Anomaly Detection Methods: Challenges and Opportunities	38
1.4.2 Chapter 3: Unsupervised Labelling with Uncertainty Estimation as a Prerequisite for Flight Anomaly Detection	39
1.4.3 Chapter 4: Ensemble Anomaly Detection for Aircraft Operations	40
1.4.4 Chapter 5: Case-Based Reasoning Approach for detecting deviations from Standard Operating Procedure in Flight Operations.....	41
1.4.5 Chapter 6: Framework for Anomaly Detection of Standard Operation Procedure (FAD-SOP) deviation.....	41
1.5 Publications status	42
References.....	42
2 A Survey of Flight Anomaly Detection Methods: Challenges and Opportunities.....	44

2.1 Introduction	44
2.1.1 The Flight Data Recorder – Data Variability	46
2.1.2 Flight Data Analysis	49
2.2 Review of Flight Anomaly Detection Methods	50
2.2.1 AD Architecture	51
2.2.2 Anomaly Detection Applications and Techniques	52
2.2.3 Flight Anomaly Detection Selection Methodology Error! Bookmark not defined.	
2.3 Flight AD limitations and opportunities	56
2.3.1 Hybrid AD Opportunity	58
2.3.2 AD-SOP Opportunity	60
2.4 Conclusion	63
References	64
3 Unsupervised Labelling with Uncertainty Estimation as a Prerequisite for Flight Anomaly Detection	69
3.1 Introduction	69

3.1.1 Flight Anomaly Detection.....	70
3.1.2 Supervised Labelling Problems	72
3.1.3 Unsupervised Labelling Solutions.....	73
3.1.4 Uncertainty Estimation.....	73
3.2 Related Work	75
3.2.1 Labelling	75
3.2.2 Active Learning (AL)	76
3.2.3 Visual-Interactive Labelling.....	77
3.2.4 Euclidean Distance Similarity Measure	78
3.2.5 Uncertainty Estimation.....	80
3.3 The Labelling Framework.....	82
3.3.1 Data and Pre-Processing	83
3.3.2 Labelling Method Selection using NASA Dataset	87
3.3.3 Labelling Results and Discussions	98

3.4 Uncertainty Estimation	107
3.4.1 Uncertainty Region	107
3.4.2 Uncertainty Estimation Results and Discussion.....	110
3.5 Conclusion	113
References.....	114
4 Ensemble Anomaly Detection Framework for Aircraft Operations.....	118
4.1 Introduction	119
4.2 Related Work	121
4.3 Ensemble Anomaly Detection Framework	124
4.3.1 Overview of Framework.....	124
4.3.2 Dataset and Data Preparation	125
4.4 Individual detectors	127
4.4.1 Individual detector: MKAD as baseline	129
4.4.2 Individual detector: ClusterAD-Sample.....	130

4.4.3 Individual detector: ClusterAD-Continuous (CAD-Cont)	132
4.4.4 Individual detector: ClusterAD Categorical (CAD-Cat)	134
4.4.5 Individual detector: CAD_ScoreMerge	137
4.4.6 Individual detector: CAD-Aggr	138
4.4.7 Individual detector: Image Processing	138
4.5 A case for segregating datatypes for CLUSTERAD	140
4.6 Evaluation: Individual Models.....	143
4.6.1 Statistical Analysis of individual methods	144
4.6.2 Contextual Analysis of Individual Methods	148
4.7 Ensemble Techniques.....	160
4.7.1 Logic-OR	161
4.7.2 Majority Voting.....	161
4.7.3 Weighting by Judgement	161
4.7.4 Weighted Majority.....	162

4.8 Evaluation: Ensemble AD	162
4.8.1 Evaluation Study III: Comparison of Ensemble Techniques	163
4.8.2 Evaluation IV: Comparison of Best Ensemble with MKAD	164
4.9 McNemar (chi-squared) Statistical Comparison: MKAD and Logic-OR Ensemble.	165
4.10 Conclusion	171
References.....	172
5 Case-Based Reasoning framework for isolating deviations from Standard Operating Procedure (SOP-CBR) in Flight Operations	175
5.1 Introduction	175
5.2 Related Work	177
5.2.1 Standard Operating Procedures	177
5.2.2 Existing Methods	178
5.2.3 A case for Case-Based Reasoning.....	179
5.3 SOP deviation detection framework with Case-Based Reasoning (SOP-CBR)	182
5.4 Demonstration Use cases	184

5.4.1 Scenario 1: A normal flight profile	184
5.4.2 Scenario 2: An Abnormal Flight with ambiguity	185
5.4.3 Scenario 3: Abnormal Flight with older case use case examples	187
5.5 Case-Based Reasoning for Statistics	189
5.5.1 The number of flight cases that constitute a specific anomaly type	190
5.5.2 Anomaly types that were typical to a particular flight crew.	194
5.6 SOP-CBR Benefits and Limitations	195
5.7 Conclusion	199
References	200
6 Discussion & Summary	203
6.1 Contributions to Knowledge addressing Research Aim & Objectives	205
6.1.1 Data Pre-processing	206
6.1.2 Labelling Framework	208
6.1.3 Ensemble Anomaly Detection	209

6.1.4 SOP-CBR	Error! Bookmark not defined.
6.1.5 FAD-SOP Demonstrations	211
6.2 Research Assumptions	211
6.3 Evaluation of FAD-SOP	212
6.3.1 Benefits of using the framework	212
6.3.2 Cons of using the Framework.....	213
6.3.3 FAD-SOP Future work.....	215
6.3.4 FAD-SOP Implications:.....	214
6.3.5 Challenges/Limitations	213
6.3.6 FAD-SOP Procedure deviation categories	216
6.4 Conclusion	217
References.....	218
APPENDICES	221
Appendix 1 – Python code for filtering for the 90-second landing window	221

Appendix II – Python Code for Creating Reduced dataset for CAD-Cat detector	223
Appendix III – Python Code to Create PLA reduced data set for CAD-Cont detector ..	226
Appendix IV – Median-based Euclidean distance Measure for one Parameter (Radio_Altitude)	227
Appendix V – Python code for main ClusterAD-DataSample Implementation	228
Appendix VI – MATLAB code for MKAD detector	236
Appendix VII – MATLAB code for Image Processing Steps.....	242
Appendix VIII – Python/MATLAB Code for McNemar’s Test.....	244

LIST OF FIGURES

Figure 1-1 Accident Rates and Onboard Fatalities by Year 1959-2020 [3]..... 26

Figure 1-2 Distribution of Two major causes of Accidents in Boeing Report [4] **Error!**
Bookmark not defined.

Figure 1-3 Fatal Accidents and Fatalities by Phase of Flight Reported by Boeing 27

Figure 1-4 Methodology flow diagram..... 34

Figure 2-1 Accident rates and fatalities by year 45

Figure 2-2 FDR formats and flow (adapted from [10]) 47

Figure 2-3 Sample of a flight dataset structure showing the varied nature of a DAR data instance
..... 48

Figure 2-4 General AD Architecture..... 51

Figure 2-5 Overview of Flight AD Technique Selection Methodology 58

Figure 2-6 Overview of Hybrid Flight AD Approach 60

Figure 2-7 Summary of flight AD applications and opportunity 61

Figure 4-1 Proposed Ensemble Anomaly Detection Framework for Flight Operations.... 125

Figure 4-2 Example of (left) ROC and (right) CDF plots for determining optimal threshold point
for detector model..... 128

Figure 4-3 ClusterAD-Sample cluster analysis representation [7] 131

Figure 4-4 Plot containing original and PLA-reduced data for CAD-Cont 133

Figure 4-5 (left) ROC and (right) CDF showing CAD-Cont score distribution with optimal threshold	133
Figure 4-6 (left) ROC and (right) CDF plot information for CAD-Cat also showing also the estimated threshold.....	137
Figure 4-7 (left) ROC and (right) CDF plot for CADScoreMerge.....	138
Figure 4-8 (left) ROC and (right) CDF plots for ImageCorr with optimal threshold	140
Figure 4-9 (top) Profile of GearRight, GearLeft and anomalous event in GearNose; (bottom) Normal profile of GearRight, GearLeft and GearNose	142
Figure 4-10 An example showing Statistical metrics described	146
Figure 4-11 (top) shows feature profiles of anomalous flight 249714. Figure (bottom) shows normal profile for selected features.....	150
Figure 4-12 Plot of abnormal flight 153633 with double inputs from both FO and Captain	152
Figure 4-13 (left) Plot of Pitch Angle in abnormal state. (right) Pitch Angle in normal state	152
Figure 4-14 (left) High Windspeed (right) Normal windspeed	152
Figure 4-15 Plot of Slat position and Altitude for a flight of interest	153
Figure 4-16 MKAD output providing insight into unusual sequence of actions	154
Figure 4-17 (top) Abnormal Roll angle and position profile (bottom) Roll Angle Profile of a normal flight	155
Figure 4-18 Instantaneous anomaly in both Autopilot features.....	156
Figure 4-19 (left) High ground speed profile (right) normal ground speed profile for majority of flights	156

Figure 4-20 Extract of Standard Procedure for final approach available to crew of the airline	157
Figure 4-21 Anomalous occurrence in sequence of action and autopilot modes.....	158
Figure 4-22 Flight 249714 with anomalous event found by ImageCorr for events detected in throttleAngle, ATS(Autothrust) and engReverser.....	160
Figure 4-23 Anomaly not identified by ImageCorr due to event only in NoseGear feature.....	160
Figure 4-24 Flights 105815 and 106201 respectively are normal flight profiles	160
Figure 5-1 Percentage of hull-loss accidents with suggested prevention strategy [1].....	176
Figure 5-2 Case-Based Reasoning Process Flow (adapted from [12]).....	180
Figure 5-3 Overview of SOP-CBR Framework	182
Figure 5-4 Normal Profile of Main and Nose Gear features covering final approach to post touchdown	187
Figure 5-5 Abnormal Profile of Main and Nose Gear features covering final approach to post touchdown for 58643	187
Figure 5-6 Extract of A320 aircraft procedures from the airline	188
Figure 5-7 (left) <i>CaptSSPitch</i> (right) <i>CaptRollSS</i> for anomalous flight.....	189
Figure 5-8 Comparing Pitch side stick inputs for captain and first officer	191
Figure 5-9 Comparing Roll side stick inputs for captain and first officer	192
Figure 5-10 (left) <i>CaptSSPitch</i> and (right) <i>CaptRollSS</i> profile also showing similar nose down inputs after Touchdown in rollout	195

Figure 6-1 FAD-SOP General Overview **Error! Bookmark not defined.**

Figure 6-2 (left) Airspeed (middle) Ground speed (right) Radio Altitude profiles for anomaly case
..... **Error! Bookmark not defined.**

Figure 6-3 (left) Speed brake mode (right) Ground spoiler mode**Error! Bookmark not defined.**

Figure 6-4 Extract of procedure for landing showing reverser guidance**Error! Bookmark not defined.**

LIST OF TABLES

Table 1-1 Research objectives to thesis chapter mapping	38
Table 2-1 Summary of flight applications and AD techniques.....	56
Table 1-1 Table of 32 continuous features in flight data	126
Table 1-2 Table of 16 categorical features in flight data	127
Table 1-3 MKAD list of anomalous flights categorised by the data type with the anomaly	130
Table 1-4 Extract from python output of the new reduced dimension of categorical features	135
Table 1-5 Example of CAD-Cat log likelihood averaged scores	136
Table 1-6 Table showing the results of all ClusterAD model variants using statistical metrics	141
Table 1-7 Table showing the results of all ClusterAD model variants using populations .	141
Table 1-8 Table showing confusion matrix description	145
Table 1-9 Summary of statistical performance of the individual detectors	147
Table 1-10 Summary of classification populations for the individual detectors	147
Table 1-11 Table summarising the evaluation tests implemented	149
Table 1-12 Summary of statistical performance of Ensemble using the different combination techniques	164
Table 1-13 Comparison of Final Ensemble performance with MKAD	165

LIST OF ABBREVIATIONS

AAIB	Air Accidents Investigation Branch
ACM	Aircraft Condition-based Maintenance
AD	Anomaly Detection
AGL	Above Ground Level
AL	Active Learning
APMS	Aviation Performance Measuring System
ASR	Air Safety Report
ATC	Air Traffic Control
CAD-Cat	ClusterAD Categorical
CAD-Cont	ClusterAD Continuous
CBR	Case-Based Reasoning
SOP-CBR	Case-Based Reasoning for Standard Operating Procedure deviation
CDF	Cumulative Distribution Function
ClusterAD	Clustering Anomaly Detection
DA	Data Analysis
DAR	Digital ACMS - Aircraft Condition Monitoring System Recorder
DBSCAN	Density-Based Spatial Clustering Algorithm with Noise

DFDR	Digital Flight Data Recorder
DMU	Data Management Unit
ED	Exceedance Detection
EDI	Edinburgh airport (EDI)
FAD-SOP	Framework for Anomaly Detection of flight crew deviation from Standard Operating Procedures
FDA	Flight Data Analysis
FDIU	Flight Data Interface Unit
FDM	Flight Data Monitoring
FDR	Flight Data Recorder
FN	False Negative
FO	First Officer
FOQA	Flight Operational Quality Assurance
FP	False Positive
FPR	False Positive Rate
GMM	Gaussian Mixture Model
GUI	Graphical User Interface
ICAO	International Civil Aviation Organisation
IMS	Inductive Monitoring System

LHR	London HeathRow airport
MKAD	Multiple Kernel Anomaly Detection
ML	Machine Learning
MV	Majority Voting
NASA	National Aeronautics and Space Administration
nLCS	normalised Longest Common Subsequence
NTSB	National Transportation Safety Board
OCSVM	One-Class Support Vector Machine
Pdf	probability density curve
QAR	Quick Access Recorder
ROC	Receivers Operating Curve
SOP	Standard Operating Procedures
SSFDR	Solid-State Flight Data Recorder
TD	Touchdown
TN	True Negative
TP	True Positive
TPR	True Positive Rate
VIL	Visual Interactive Learning

VL	Visual Learning
WJ	Weighted Judgment
WMV	Weighted Majority Voting
XML	eXtensible Markup Language

1 Introduction

Air travel is deemed the safest mode of transportation in recent times. Aviation safety is measured in terms of accident rates which is the ratio between the number of reported flight accidents each year and the total number that were conducted in the same year expressed in accidents per million flights [1]. 'Accidents' in this domain, refer to aircraft operational occurrence where hurt or loss of life occurs or there is substantial damage to the aircraft. Figure 1-1 provides reported worldwide commercial fleet accident rates from 1959 till 2020 and the trend shows that although the accident rates have decreased in the last decades, fatalities still occur today.

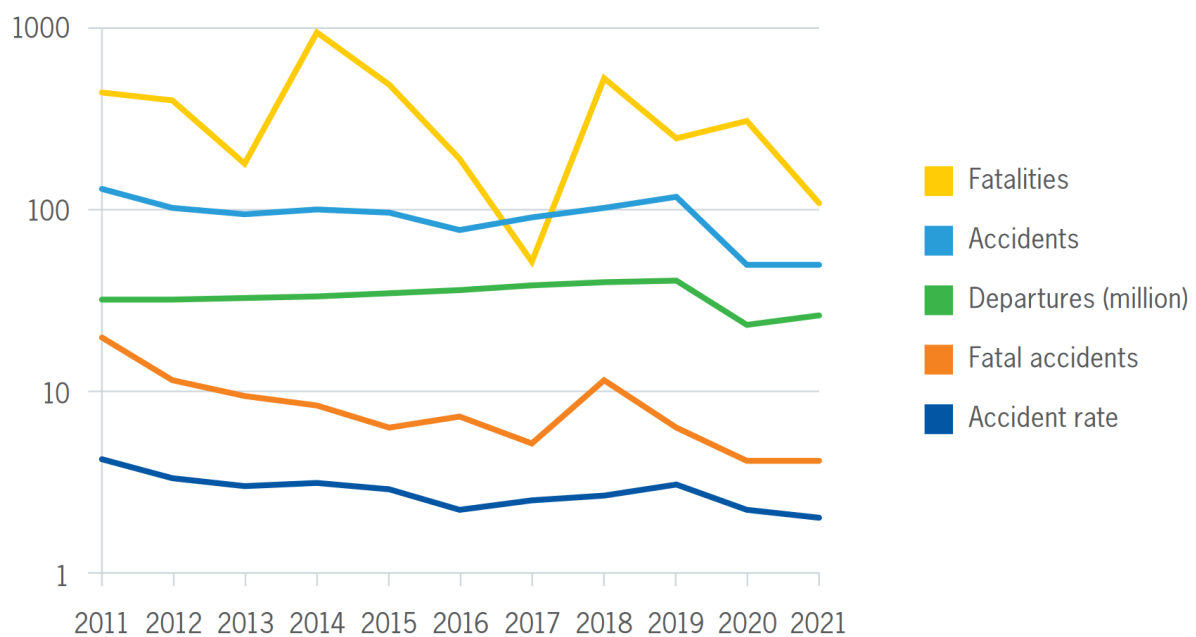


Figure 1-1 Accident Rates and Onboard Fatalities for scheduled commercial flights [3]

With a predicted increase in air traffic [2], it can be deduced from the trends that with successful safety enhancements, accidents rates can be decreased. Therefore, it comes as no surprise that Aviation safety improvement continues to be a major focus of the aviation industry.

Although the complex aircraft system can be put in an unstable state through the combination of many factors such as mechanical defects and weather problems, the industry agrees that human error accounts for about 80% of accidents [4]. The interaction of humans with the aircraft machine means that errors can emanate from mechanics, air traffic controllers, pilots, etc. However, the primary contributor to most accidents is said to be pilot errors that occur due to deviations from organisational expectations laid out in Standard Operating Procedures (SOP) [5]. Such errors range from incorrect altitude settings to failing to put down flaps at landing by the flight crew. Most safety related improvements over the past few decades have focused on taxiing, climbing, approach, and landing phases. Figure 1-2 reveals the reason for this to be that fatal accidents occur most during these phases particularly the final approach and landing which accounts for over half of all the fatalities. Deviations from SOP are more pronounced in phases of flight with a higher pilot activity requirement such as the final approach and landing phases.

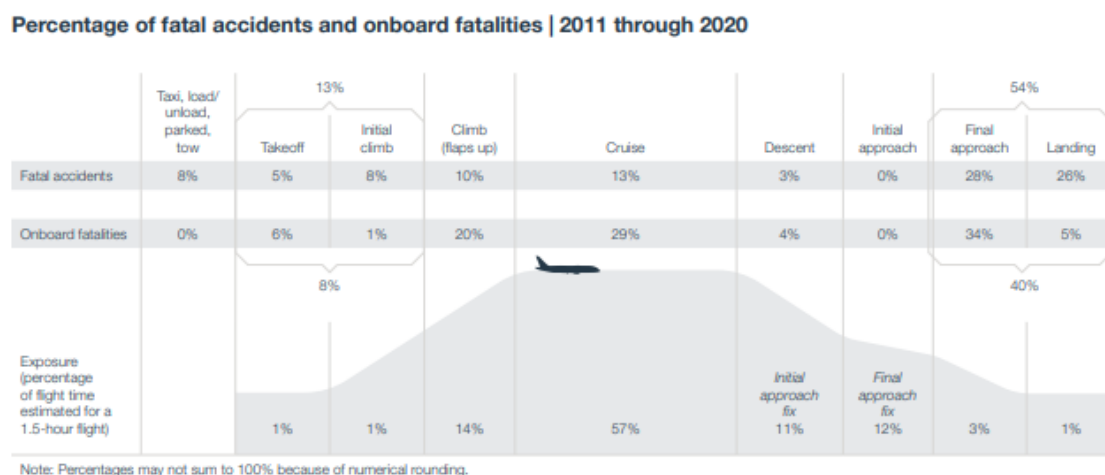


Figure 1-2 Fatal Accidents and Fatalities by Phase of Flight Reported by Boeing

In a bid to reduce accident rates, the airline industry has previously focused on a reactive approach to identifying and mitigating operational risks after accidents have occurred. However, in a safety-critical environment such as this, there is a requirement for more proactive approaches that detect abnormalities and potentially hazardous events before serious incidents or accidents occur. Several bodies within the industry have come up with different voluntary programs for proactive system management. They have since adopted the use of Flight Operational Quality Assurance (FOQA) also

known as Flight Data Monitoring (FDM) to collect, extract and analyse digital data from various aircraft sub systems during flight operation. Substantial progress has been made in identifying precursor events, information on preferred operational processes being adopted by flight crew and assessing risks, all with a view to putting in place mitigation steps to prevent related incidents from happening in subsequent flights.

1.1 Research Motivation

Despite the existing efforts to improve airline safety in reducing fatalities and other related accident costs, there are still serious incidents and accidents still being reported today revealing the need for even better safety enhancement approaches. Figure 1-1 confirms a steady increase in accident rates between 2017 and 2019 before the global pandemic resulted in a sharp decline of accidents due to restricted air travel during the period. In essence, the use of traditional FDM approaches adopted within the industry today cater only for known incidents found in the flight data including known procedure deviations while missing unknown events or combination of events. In this accident [6], the flight crew failed to use an existing abnormal operating procedure that was available for dealing with the situation. Also, investigations revealed that prior cases with similar inflight events had been documented by several pilots and not picked up in flight data analysis. This situation suggests that the industry will benefit from finding such previously unknown and underlying procedure confusions in flight data before they lead to accidents and forms the rationale behind this research.

1.1.1 Flight Data

Flight investigations and analysis often combine information from various sources such as cockpit voice recorders, maintenance logs, air traffic controller information and digital flight data. The latter is obtained from the aircraft's digital sub systems and transported to Flight Data Recorders (FDR) where the individual system information is integrated thereby providing analysts with the comprehensive data they need. In the literature review, details about the evolution of flight data are presented. This data adds to insight from the manual reports given by flight crew to provide a wholistic

picture of flight operations. As an example, a hard landing would produce a quantitative perspective in an FDM event, context from the flight crew report as well as the results from an airworthiness report.

In this work, data from FDR are used because of its volume and varied nature as it contains huge amounts of measurements from different systems in one dataset. While some quantities are recorded as continuous data types, others are in categorical form. The airspeed, altitude and pitch angles are recorded as continuous parameters while parameters like the auto-flight and brake-deploy modes are categorical parameters. Modern aircrafts such as Airbus A350 and Boeing 787 can record thousands of these variables on their recorders covering, not just measurements, but also system state [7]. FDM relies on the reliable collection, recording and transmission of correct and appropriate data in identifying risky events.

1.1.2 Anomaly Detection

The traditional approach for FDM that identifies deviations from manual limits and SOP is referred to as Exceedance Detection (ED). It consists of a set of algorithmically coded core events that cover special areas of interest that is common among operators [8]. A hard landing and flap limit speed exceedance are examples of events that are sure to be picked up by this method. As incidents happen, the known events database can be built up by adding more events. The problem here is two-fold: first, emerging risks cannot be identified by this classic method and secondly for a new event to be added to the database, current methods are reactive being able to get such information only after an incident occurs.

This research reviews the Anomaly Detection (AD) approach which is a more proactive approach that highlights abnormal events within a flight that are different from those of the majority. This approach helps find insights within the data that help understand operational preferences of flight crew, different variations of normal, outlier flights, potential incident precursors, procedure deviations and even normalisation of deviance cases that may be evident within the flight data. The benefit of this method, in addressing the two-fold problem from exceedance detection, is that it can identify

emerging risk events and those events can then be added to the Exceedance database without the incident occurring prior.

Various AD techniques have been developed and they differ based on how they perform in detecting anomalies in a heterogeneous dataset like the flight data described. For example, some techniques work well in detecting risky events in continuous data and would miss highlighting anomalous events existing in categorical parameters. This formed the basis for proposing an ensemble AD technique that combines the strength of multiple algorithms to increase the chance of identifying more anomalous flights than individual methods. It is assumed that with a wider array of anomalies being detected, there will also be a higher chance of detecting even more of those anomalies that are due to procedure deviations which is the goal.

1.1.3 Standard Operating Procedures

The current FDM approach is used for detecting SOP deviations and so the limitations of the approach adversely affect procedure deviations it is meant to find. In other words, any emerging procedure deviation event that has not been algorithmically coded into the traditional method will be missed thereby increasing the probability of it becoming a precursor that triggers an incident. Only a few literature [9] focused on procedure deviation detection and they follow a rule-based modelling approach which has similar limitations with the ED approach. There is therefore a need to use more unsupervised means to detect subtle underlying procedure confusions of flight crew before they lead to severe outcomes.

Reasoning has been used in various applications where logic and cognition are required for problem-solving. For the aviation sector, reasoning has been used for incident reporting and maintenance troubleshooting [10]. The reasoning system and/or algorithms that perform the reasoning operation are known in literature as “reasoners”. A type of reasoning system that is used in this work is referred to as a Case-Based Reasoning (CBR) system. It depends on historical flight information to build its case knowledge database from which exact matches of new instances can be retrieved and labelled. The reasoner in this work is adapted to label a new instance as an SOP

deviation instance based on the similarity matching with other flight cases in the case database.

1.2 Aim and Objectives

Aim

To develop a **F**ramework for **A**nomaly **D**etection of **S**tandard **O**perating **P**rocedure (FAD-SOP) deviations from flight data integrating an Ensemble AD to detect wider range of anomalies and a reasoning system to highlight the anomalies that are due to procedure deviations.

Objectives

The aim can be achieved through a set of objectives ranging from finding gaps in literature to developing a framework which identifies emerging procedure deviations in flight operations using FDR data, data analytics and expert knowledge.

Objective 1:

A review of existing work and seeking out gaps in literature focusing on specific areas such as:

- i. evolution of improvement methods for Aviation safety
- ii. flight Anomaly Detection techniques for all anomaly types
- iii. existing methods tailored towards procedure deviation in flight operations
- iv. case based reasoning systems

Objective 2:

The extraction and pre-processing of the flight data as well as labelling all flight instances in the dataset to be used effectively for machine learning.

Objective 3:

To develop an Ensemble AD technique for finding a wider range of anomalies than individual detection methods.

Objective 4:

Design a Case-based reasoning module that isolates anomalies that are specific to procedure deviations in flight operations.

Objective 5:

To evaluate the ability of the FAD-SOP which integrates the robustness of an Ensemble AD with the power of reasoning to detect more anomalies, isolate those with procedure deviation risks and identify their causal factors.

1.3 Methodology

The research methodology sets the scope of work to be achieved, defines the aims and objectives, as well as timelines for achieving them. The objectives are set out to be carried out in five phases as summarised below and shown in Figure 1-3.

1.3.1 Phase 1: Literature Review

Objective 1 is achieved in this phase involving setting up the background and scope of the research work. The identification of research gaps was possible through in-depth review of literature about various related subjects. The topic areas covered in the review were aviation safety, flight data, flight anomaly detection methods, standard operating procedures in flight operations and case-based reasoning techniques. The following research gaps were identified because of work carried out in this phase.

Research Gap 1

Most aircraft accidents are agreed to be caused by pilot error and a primary contributing factor is their deviations from standard operating procedure laid out by the industry and airlines to ensure air safety. This realisation was significant in motivating and forming the problem statement for this research.

Research Gap 2

Anomaly detection has been researched over the years in flight data analytics and it has shown promising results over traditional approaches in detecting emerging risks in flight operations. However, the methods that have been developed for flight AD have been limited in the type of anomalous flights they are able to find thereby leaving room for missing potentially hazardous events.

Research Gap 3

The literature revealed that existing methods tailored towards detecting procedure deviations in aircraft operations are rule-based with an obvious limitation: difficulty in listing all possible rules. Therefore, the need for more unsupervised methods, that learn relationships from the data, are desired.

Outcome of Phase 1: In this literature review phase, context was highlighted, aims were set and objectives were defined. In addition, one conference paper was produced.

Igenewari, V. R., Skaf, Z., & Jennions, I. K. (2019, September). A Survey of Flight Anomaly Detection Methods: Challenges and Opportunities. In *Annual Conference of the PHM Society* (Vol. 11, No. 1). <https://doi.org/10.36001/phmconf.2019.v11i1.898>

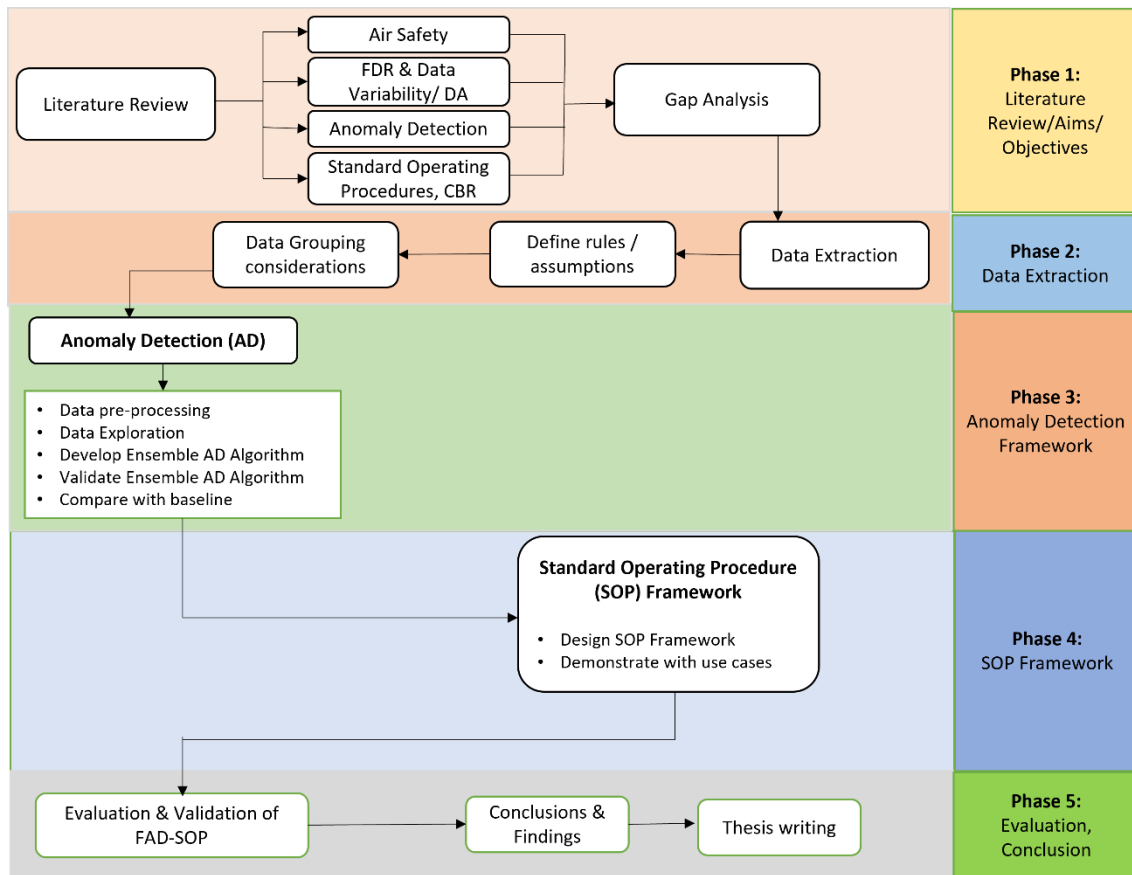


Figure 1-3 Methodology flow diagram

The subsequent methodology phases described in more details the research gaps identified and frequent literature reviews carried out throughout this work helped highlight latest methods being published in recent studies.

1.3.2 Phase 2: Data Analytics and Labelling

In this phase, **Objective 2** was met by formulating the concept behind the proposed framework, FAD-SOP. Also, the dataset for a particular aircraft fleet, flight route while focusing on a flight phase, was selected, extracted into a usable format, pre-processed, and explored. This phase was the most time-consuming as it involved establishing the reasons for choosing the dataset, ensuring high-quality data is produced after the pre-processing and the data instances are labelled as either normal

or anomalous. The labelling was a non-trivial task as it supplied the training data and evaluation information for the next phases of this research.

Outcome of Phase 2:

To proceed with the implementation of the framework, the dataset needed to be prepared and assumption made about it. Though a publicly available dataset was also used, the core of the research work was hinged on how the approach performed on a real airline dataset. In this phase, the following were produced as outcome:

- i. selection of the dataset, that is, a fleet of A320 aircrafts comprising over ten thousand flights that operated between 2002 to 2012 and flew the route from any runway at London Heathrow airport (LHR) to any runway at Edinburgh airport (EDI) while focusing on the landing phase of flight.
- ii. proposing of an unsupervised method for the initial labelling of the flight instances while measuring the level of uncertainty in the labelling approach used.
- iii. produced a journal paper which has been submitted titled:

Unsupervised Labelling with Uncertainty Estimation as a Prerequisite for Flight Anomaly Detection

1.3.3 Phase 3: Ensemble Anomaly Detection

This phase involved developing the first building block of FAD-SOP to meet **Objective 3**. As observed in Figure 1-3, it is the first part of the framework which develops and implements an ensemble AD framework comprising of a combination of individual AD techniques on the real flight dataset. While implementing the existing flight AD methods, the following research gap was found and addressed by the proposed ensemble AD solution.

Research Gap 4

Recent AD techniques tailored to aircraft operations are Multi Kernel Anomaly Detection (MKAD) and Cluster-based Anomaly Detection (ClusterAD). In previous

research, a comparison was made to ascertain the performance of these two algorithms, and it was found that while MKAD did well in identifying flights with anomalies in their categorical data parameters, it did not perform as well when anomalies exist in the continuous parameters of the flight data. The reverse was the case for ClusterAD as it detected anomalies in continuous variables but was limited in its ability to detect abnormalities in categorical variables. There is therefore a need to develop a method that combines the strengths of both techniques to avoid missing emerging risks evident in the data.

Outcome of Phase 3:

In this phase, MKAD, ClusterAD along with other methods such as Image processing and KPrototype were implemented individually as baseline techniques and then made as building blocks into an ensemble combination and the performance determined. This is how research gap 4 is tackled. A statistical as well as a contextual evaluation of the baseline and ensemble methods was observed and flight instances that revealed the similarities and differences were highlighted. The implementation also evaluated how the ensemble performed on a different flight route and gave reasons for how it is expected to perform on a different flight phase. The outcome here, is also in the form of a journal paper that has been submitted titled:

An Ensemble Anomaly Detection for Flight Operations

1.3.4 Phase 4: Standard Operating Procedure Design

Here, a case-based reasoning system is reviewed and introduced for anomalous flight detection that is specifically due to SOP deviations by the flight crew. A design is proposed to isolated procedure deviations from other type of anomalies that were found by the Ensemble AD in the previous phase.

Outcome of Phase 4:

A review of the reasoning methods is carried out, the reasoning architectural design is described, and demonstration use cases are used to explain its capability in this research. Use cases observed during implementation are used to demonstrate the

flow from the ensemble building block of the framework to the reasoning stage. It reiterates the outcome of the Ensemble phase and how the procedure deviations can be isolated from the wider range of anomalies flights discovered by the ensemble. The work carried out is ready to be submitted for publication. It is titled:

A Case-Based Reasoning Approach to detecting deviations from Standard Operating Procedure in Flight Operations

1.3.5 Phase 5: FAD-SOP Evaluation and Conclusion

In this phase, FAD-SOP is evaluated and compared with other work to showcase its strengths. Future research areas to which this work can be extended are also explored in this phase.

1.4 Thesis Structure

A 'Paper-based' format is adopted for writing this thesis consisting of six chapters. Apart from chapter 1 which introduces the research structure, and chapter 6 which concludes the research work, all other chapters have either been published or have been submitted to be peer-reviewed. Chapter 3, 4 and 5 have been submitted to journals. The paper-based chapters have been re-formatted within the thesis to ensure the research journey is clearer to the reader and they present the outcomes of different phases of the research methodology. The thesis structure seeks to make the achievement of the objectives clear within the chapter arrangement and Table 1-1 shows the objective to chapter mapping.

Table 1-1 Research objectives to thesis chapter mapping

Chapter	Methodology Phase	Objectives	Publication Status
Chapter 2	Phase 1	Objective 1: Literature Review	Published in conference
Chapter 3	Phase 2	Objective 2: Data Labelling	Submitted to journal
Chapter 4	Phase 3	Objective 3: Ensemble AD	Submitted to journal
Chapter 5	Phase 4	Objective 4: SOP-CBR	Submitted to journal

The following subsections summarise the content of each chapter in this thesis.

1.4.1 Chapter 2: A survey of Flight Anomaly Detection Methods: Challenges and Opportunities

This chapter is an outcome of **Phase 1** of the research methodology and achieves **Objective 1**. It is the chapter that reviews existing work within the scope of this research and summarises the gaps found. These gaps will be described in more detail within the relevant chapters and the proposed solutions introduced. Chapter 2 introduces the field of AD, contains a description of the Flight Data Recorder (FDR) as the data source for this work, reviews the existing techniques for flight AD and the aviation applications in which they are used. In addition, the chapter touches on the criteria for selecting appropriate flight AD methods, their limitations and the suggests an opportunity to be explored in using AD for SOP deviation detection.

Chapter 2 Outcomes:

- i. This chapter introduces FDR particularly data variability as a key factor for the continued development on flight AD methods.

- ii. The evolution of flight AD techniques is explored, and their individual limitations identified as one of the research gaps for this work.
- iii. An opportunity for applying AD to the detection of procedure deviations in aircraft operations is suggested.

All these are covered in the following publication and reformatted into chapter 2. It is worthy of note that in this publication, the term ‘hybrid’ was used. However, more reviews revealed the difference in definition with the concept of an ‘ensemble’. Therefore, this work after that publication made use of the term ‘ensemble’ to qualify the integration of multiple data-driven methods as opposed to the likely confusion with a combination of data-driven and model-based methods to which the term ‘hybrid’ mostly refers.

Igenewari, V. R., Skaf, Z., & Jennions, I. K. (2019, September). A Survey of Flight Anomaly Detection Methods: Challenges and Opportunities. In *Annual Conference of the PHM Society* (Vol. 11, No. 1). <https://doi.org/10.36001/phmconf.2019.v11i1.898>

1.4.2 Chapter 3: Unsupervised Labelling with Uncertainty Estimation as a Prerequisite for Flight Anomaly Detection

This chapter aligns with Objective 2 and is an outcome of Phase 2 of the methodology workflow. This chapter describes the datasets used in this research and the implementation of more unsupervised techniques that reduce the cost of manually labelling data while ensuring high accuracy of the approach. Also implemented in this work is estimating how sure the labelling approach is considering that the labels it produces would be input into the next FAD-SOP module.

Chapter 3 Outcomes:

- i. A description of the proposed approach for labelling flight data
- ii. The evaluation results of the labelling approach
- iii. The degree of uncertainty that exists with the approach

This chapter forms a reformatted version of the following journal paper submitted for publication:

Unsupervised Labelling with Uncertainty Estimation as a Prerequisite for Flight Anomaly Detection

1.4.3 Chapter 4: Ensemble Anomaly Detection for Aircraft Operations

Chapter four seeks to achieve **Objective 3** and aligns with **Phase 3** of the research methodology. It is aimed at describing the characteristics of the baseline anomaly detector techniques, justify their usage, implement on flight dataset, and analyse their individual performance. It also details how the outcome of the labelling module serves as input into the implementation in this chapter contributing to both training and evaluation of the methods. The chapter then makes a case for the ensemble AD proposal, describes the approach, evaluates its performance against the ground truth labels from Phase 2 and compares the statistical and contextual performance with the baseline methods while drawing logical conclusions.

Chapter 4 Outcomes:

- i. Detailed description, implementation and performance of individual methods adopted for detecting flight anomalies
- ii. Introduction of ensemble methods implemented and the selection of the best performing approach
- iii. Statistical and contextual discussion of the advantage of the ensemble over the individual methods

Chapter 4 is produced from the following journal paper submitted for publication:

1.4.4 Chapter 5: Case-Based Reasoning Approach for detecting deviations from Standard Operating Procedure in Flight Operations

Chapter 5 aligns with **Phase 4** of the research methodology workflow and achieves **Objective 4**. This chapter is aimed at isolating procedure deviations from the wide array of anomalies detected in Phase 3 of the research. It begins with definitions and review of existing procedure deviation methods as well as related reasoning techniques and applications. A reasoning design is proposed and demonstrated through a thought experiment using example use cases.

Chapter 5 Outcomes:

- i. Review of SOP methods for flight anomaly detection
- ii. Review of CBR approaches and applications
- iii. A CBR approach to SOP deviation detection design is demonstrated using use case examples.
- iv. A visualisation GUI to aid expert decision making

Chapter 5 presents the content of the following journal paper available for publication:

A Case-Based Reasoning Approach to detecting deviations from Standard Operating Procedure in Flight Operations

1.4.5 Chapter 6: Framework for Anomaly Detection of Standard Operation Procedure (FAD-SOP) deviation

This chapter pulls all the previous chapters together, describing the implementation, specifically combining a developed module with a conceptual design experiment with real cases found in the data. Chapter 6 is mapped with **Phase 5** of the research methodology and achieves **Objective 5**. This chapter also summarises the thesis, showcases the contributions of the research, discusses the limitations and future directions that can be explored.

Chapter 6 Outcomes:

- i. Contribution to Knowledge addressing research objectives
- ii. Research assumptions
- iii. Pros, Cons, Implications, Challenges, Future Work of FAD-SOP
- iv. FAD-SOP procedure deviation categories observed in this study

1.5 Publications status

Conference Papers (published)

- i. Igenewari, V. R., Skaf, Z., & Jennions, I. K. (2019, September). A Survey of Flight Anomaly Detection Methods: Challenges and Opportunities. In *Annual Conference of the PHM Society* (Vol. 11, No. 1). <https://doi.org/10.36001/phmconf.2019.v11i1.898>

Journal Papers (submitted)

- i. *Unsupervised Labelling with Uncertainty Estimation of Flight Data*
- ii. *An Ensemble Anomaly Detection for Flight Operations*
- iii. *A Case-Based Reasoning Approach to detecting deviations from Standard Operating Procedure in Flight Operations*

References

1. *ICAO Methodology for Accident Rate Calculation and Trending*. Safety Regulations > Monitoring and Oversight, 2021.
2. Airbus, *A Statistical Analysis of Commercial Aviation Accidents 1958-2018*. 2019, Airbus: France.
3. https://www.icao.int/safety/Documents/ICAO_SR_2022.pdf
4. Wiegmann, D.A., & Shappell, S.A. (2003). *A Human Error Approach to Aviation Accident Analysis: The Human Factors Analysis and Classification System* (1st ed.). Routledge. <https://doi.org/10.4324/9781315263878>.

5. Sumwalt, R.L. *Standard Operating Procedures: The backbone of professional flight operations*. 2013 [cited 2020 12.12.2020]
6. BEA, *On the accident on 1st June 2009 to the Airbus A330-203 registered F-GZCP operated by Air France flight AF 447 Rio de Janeiro - Paris*, in *Safety Investigations*. 2012, Bureau d'Enquêtes et d'Analyses: France.
7. Barry, D.J., *Annotated 'normal' flight instances can be used as a yardstick for finding anomalous flights.*, in *School of Aerospace, Transport & Manufacturing*. 2019, Cranfield University. p. 263.
8. (CAA), C.C.A.A., *Flight Data Monitoring*. 2013, CAA: UK.
9. Maille, N. *Modeling airline crew activity to improve flight safety analysis*. in *2017 IEEE Aerospace Conference*. 2017. Big Sky, MT, USA: IEEE.
10. Ezhilarasu, C.M., Z. Skaf, and I.K. Jennions, *The application of reasoning to aerospace Integrated Vehicle Health Management (IVHM): Challenges and opportunities*. *Progress in aerospace sciences*, 2019. **105**: p. 60-73.

2 A Survey of Flight Anomaly Detection Methods: Challenges and Opportunities

Safety enhancement is a major goal of the aviation industry owing to the predicted increase in air travel. There is also the need to prevent fatalities, increase reliability and reduce monetary costs suffered because of delays and accidents that still occur. Accidents today are complex because of many causal factors acting alone but more often as a combination with other contributing factors. In tackling this trend, proactive measures have been put in place to find hazardous combinations that occur during flights to mitigate them before accidents occur. Flight Anomaly Detection (AD) methods are aimed at highlighting abnormal occurrences of a flight, that are different from the norm. As an improvement on the current state-of-the-art method, previous works have proposed different AD techniques for detection of previously unknown flight risks such as component faults, aircraft operational inefficiencies and some abnormal crew behaviour. However, current AD methods individually have limitations that prevent them from detecting certain significant anomalies in flight data. This paper surveys current flight AD approaches, their strengths and limitations as well as brings to light the benefits of a hybrid AD method. It extends previous work and finds safety-critical events, particularly those related to abnormal crew activity: a class of events known to amount for a substantial number of accidents/incidents today. It also highlights another emerging AD application opportunity, its challenges and how AD is beneficial in addressing them.

2.1 Introduction

The sharp decline in aviation accident rates from 5 to 0.22 per million departures [1] over the last 50 years is the reason air travel is deemed the safest means of transportation today as shown in Figure 2-1Figure 1-1. However, accidents and serious incidents still occur which affect the reputation of the industry. An **Accident** refers to an occurrence associated with the operation of an aircraft in which any person suffers death or serious injury, or in which the aircraft receives substantial damage. An

Incident is "an occurrence, other than an accident, associated with the operation of an aircraft which affects or could affect the safety of operation." [2]. Examples of incidents are high-speed approach, improper flap configuration, controlled flight into terrain only marginally avoided, aborted take-offs/landings on a closed or engaged runway. Improper actions often lead to a hazardous system state which in turn lead to an incident effect and if not addressed in time, will eventually lead to accidents.

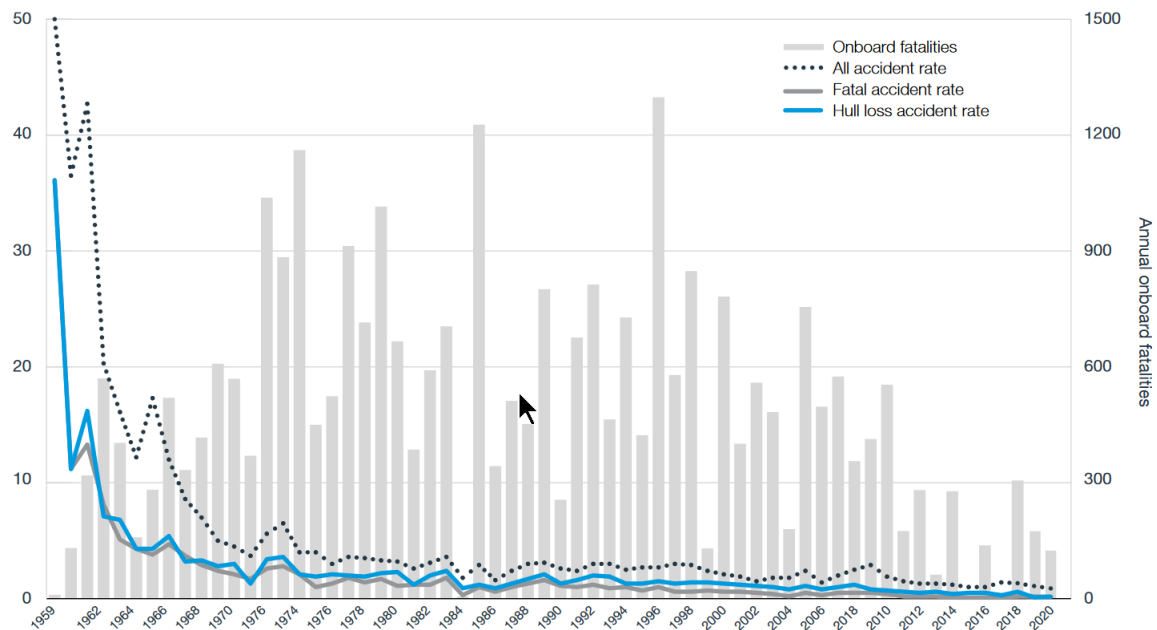


Figure 2-1 Accident rates and onboard fatalities by year [2]

In addition to the loss of lives involved in air accidents, airlines suffer huge monetary costs as a result of aircraft damage, delays, and other accident-related costs [3]. For example, Boeing publicises it costs approximately \$300 million to buy one of the newer commercial aircraft models [4] and no less than \$10,000 per hour may be expended for any delay in getting the aircraft airborne [5]. With the growing demand for air travel [6], there is a possibility that accident numbers could increase if the safety levels are not enhanced simultaneously. According to the International Civil Aviation Organisation (ICAO) which is responsible for providing the standards for aviation safety, performance monitoring of the aircraft systems, maintenance operations, as well as the monitoring of the flight crew are some of the plans in place to achieve the desired safety level [7]. This monitoring aims to detect risks well in advance of serious accidents thereby preventing loss of lives and lowering the financial losses in the

aviation sector [8]. Within the aerospace industry, an aircraft is able to gather up to 300,000 parameters from its sensors, depending on aircraft type and flight hours [9]. A practical example in [5] shows that a cross- country flight in the United States is able to produce 240 terabytes of data from both engines of a Boeing 737 aircraft after a 6-hour flight. This size of data increases rapidly depending on the number of aircraft in the sky and for say, a year. Such data is referred to as “Big Data” and the above example shows that the aerospace industry understands more than other industries, the challenges involved in making use of technologies to extract value from these huge volumes of data. Big data is often in varied formats that change at a high velocity and vary in context. Data Analytics (DA) is the science that acts on Big Data by using automated algorithms to uncover hidden patterns and insights for better decision making. In the aerospace industry, data analytics can be generally useful for cost reduction and revenue maximization which is achieved through means like better maintenance scheduling, performance monitoring and optimization of resources. Performance monitoring of systems, processes and people has become essential to safety in the aerospace industry [7].

Improvements made to enhance safety have mostly been reactive as a response to accidents that have occurred. Flight Data Monitoring (FDM) programmes were introduced as a proactive approach for gathering and analysing routine flight data to improve flight crew performance, operating procedures as well as aid pilot training. These programmes were aimed at identifying hazards before they led to accidents thereby aiding the establishment of appropriate mitigation strategies to prevent such occurrences. Data gathering for FDM programmes are available from onboard aircraft recorders, voice systems, maintenance reports, safety reports, cockpit display messages, or weather systems. The onboard data recorder sources, however, produce the “Big Data” mostly useful for the analysis phase of FDM.

2.1.1 The Flight Data Recorder – Data Variability

The Flight Data Recorder (FDR) provides access to aircraft system data with which data analytics techniques extract the information required to carry out their processes. Figure 2-2 shows the flow of data to and from the FDR. All monitored aircraft systems

pass their measurements through the information bus for the Flight Data Interface Unit (FDIU) and Data Management Unit (DMU) to select the parameters they have been programmed to process. Different recording units such as the current Solid-State Flight Data Recorder (SSFDR) and the Digital FDRs (DFDRs) store the appropriate parameters and produce different data frames which in turn require decoding programs to output useful data formats for analysis.

The SSFDR has expanded recording capacity and improved recorder reliability as an improvement to the DFDR which in turn replaced the older magnetic tape versions. The FDIU provides the required interface between the various sensors and the DFDR thereby solving the problem of the DFDRs' inability to process the larger amounts of incoming sensor data. It converts analogue signals from the sensors to digital signals suitable for recording by the DFDR. However, the FDIU is not programmable by the operator as it sends only basic regulatory information to the DFDR/SSFDR. The DMU, on the other hand, can be programmed by an operator to record any convenient parameters ultimately producing the DAR (Digital ACMS - Aircraft Condition Monitoring System Recorder) data as also shown in Figure 2-2.

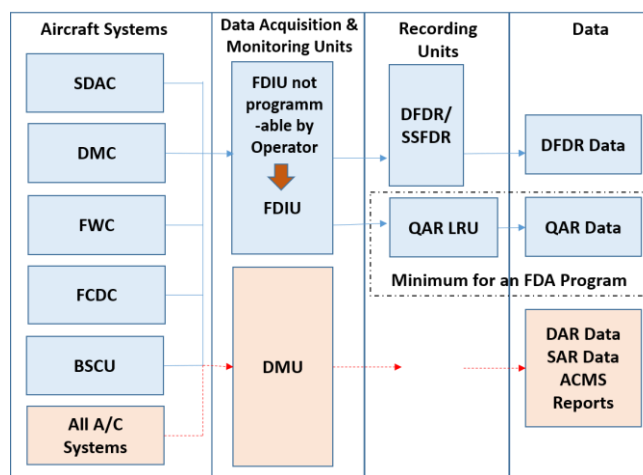


Figure 2-2 FDR formats and flow (adapted from [10])

For analysis purposes, both the QAR (Quick Access Recorder) and DAR data are used based on ease of information retrieval and/or variability of parameters they can provide. The QAR is a copy of the FDR on a non-crash resistant recorder which has the specificity to be easy to download and interchangeable [11]. DARs are like QARs

[10] however, while the QAR data originates from the FDIU with minimum regulatory parameters as the FDR, the DAR data, on the other hand, is the product of the DMU able to include many more convenient parameters. Both systems were developed to ease data recovery for the airlines' needs and contain high-frequency sensor data.

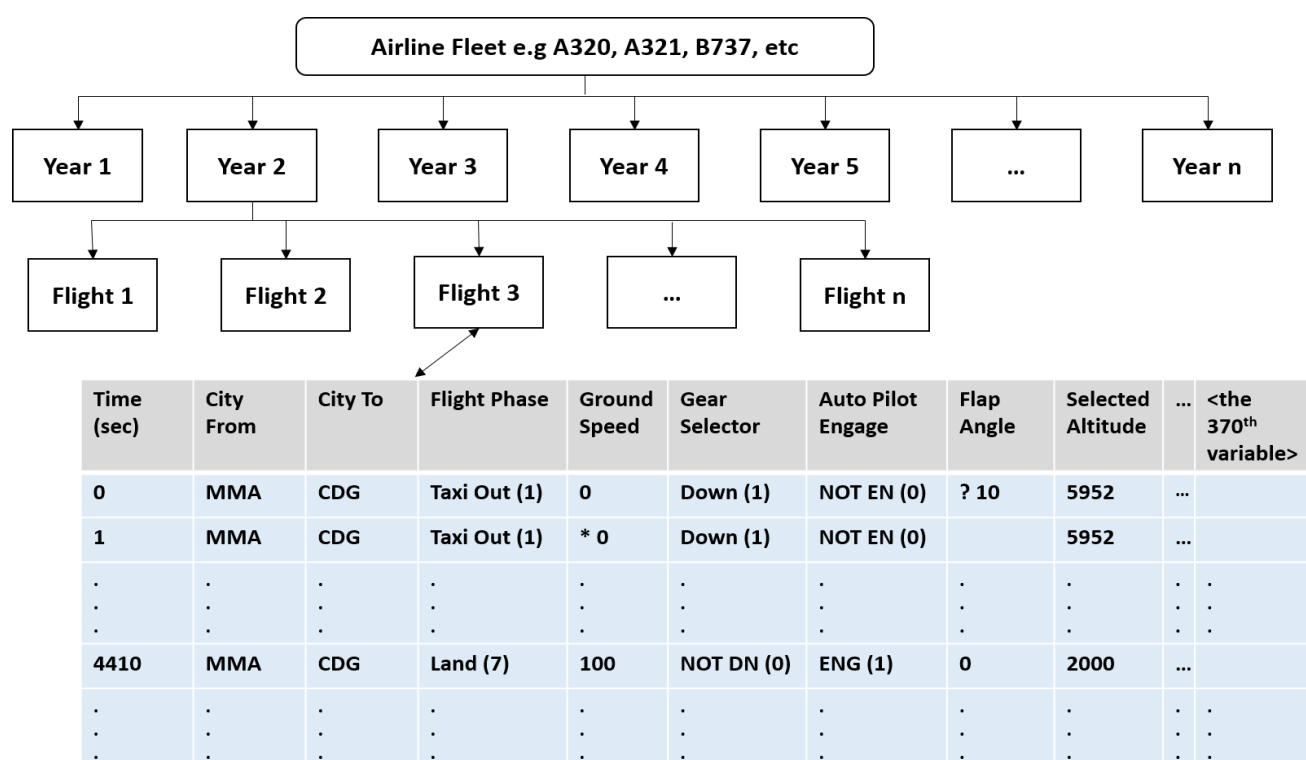


Figure 2-3 Sample of a flight dataset structure showing the varied nature of a DAR data instance

These different aircraft devices record parameters of different volumes, different data types and at varied speeds thereby giving rise to multiple data dimensions and data inconsistencies among other challenges of flight data analytics. Figure 2-3 shows an example of the variable nature of flight data. It reveals particularly, the varied nature of FDR data such as the presence of **discrete variables** like the landing gear state and autopilot modes as well as **continuous variables** like altitude, ground speed and so on. The example in this figure is a DAR data set from an airline's fleet of aircrafts for several years with each year consisting of numerous flight instances. Each raw flight instance data holds about 370 parameters, 5000 samples of parameter values recorded at different speeds and inconsistencies such as missing values, noise

values, incorrect values among others. Though the varied nature of the parameters that recorders are now able to record can provide more insights into occurrences during a flight, data-driven AD techniques must deal with these data variability challenges in various ways to achieve desired results.

The following sections reveal the effectiveness of current and emerging flight data analytics techniques, their strengths, and weaknesses with regards to detecting anomalies while dealing with data variability from FDR.

2.1.2 Flight Data Analysis

The success of FDM depends on insights that can be gained from existing flight data through Flight Data Analysis (FDA) also known as Flight Operations Quality Assurance (FOQA) in the improvement of aviation safety. The effective analysis of flight data has been enhanced by the increased number of parameters that can be measured and the emerging techniques in data analytics.

Exceedance Detection (ED) is one of such data analytics techniques. It is the state-of-the-art method used across the industry for flight data analysis. ED is implemented as a knowledge-based technique which relies on input from domain experts to make its inference within the data. It is, therefore, only able to highlight known risks within the data. More recent studies have built on the limitation of the ED method giving rise to the Anomaly Detection (AD) approach aimed at identifying unknown abnormalities during the flight which cannot be detected by the current method. Highlighting unknown combinations within the aircraft environment is useful in detecting operationally significant events that could potentially lead to accidents. Anomaly Detection approaches could either be **Model-based** relying on domain knowledge to build models or **Data-Driven** which focuses on automatically learning the general rules to build models directly from the data. However, for a complex system like the aircraft, it is not possible to model every combination of factors using model-based methods. Therefore, the advancement in technologies like data mining, data analytics and machine learning are moving the focus from knowledge-based approaches to data-driven methods of improving FDA.

Data-driven approaches are used to describe system behaviour using only the data. Data-driven methods for data analysis exploits large amounts of data available to better understand behaviours and operational faults. These approaches learn by observing and analysing system behaviour of complex systems. The goal here is to build systems that can observe, diagnose, recognise unusual events and inform operators for more effective decision making [12].

This paper thus explores existing data-driven flight AD techniques, their applications, strengths, limitations and proposes a more robust hybrid AD technique to detect a wider range of emerging significant anomalies aimed at enhancing flight safety. The paper is organised as follows. An introduction of FDM, the FDR data variability and FDA are provided in this section. Section 2.2 reviews the flight AD evolution, its architecture, aviation applications and its methods as well as the strengths and limitations of these methods. The section also gives an overview of a method of selecting appropriate flight AD techniques. Section 2.3 summarises limitations in existing flight AD techniques and proposes a hybrid AD opportunity to mitigate some of them. Also, the application of AD to the area of Standard Operating Procedure (SOP) deviation detection was introduced as another opportunity. Lastly, section 2.4 concludes the paper.

2.2 Review of Flight Anomaly Detection Methods

Anomaly Detection refers to the task of identifying new or unknown patterns which, in many cases are abnormal or inconsistent from the norm. AD finds patterns in data that do not conform to expected behaviour [13]. These patterns are called anomalies or outliers and they are often sufficiently different from most data points in the dataset. Since the aircraft operations are standardized, it is assumed that most flights represent normal operations at that given time. Flight data analysis has given rise to several AD techniques for discovering anomalies.

This section details the general architecture of an AD method. It also provides a review of current flight AD application areas, the different AD techniques developed for use in each application as well as the strengths and limitations of these techniques.

2.2.1 AD Architecture

Each AD technique has a slightly different way of approaching the AD problem depending on available data and anomalous behaviour they wish to detect. The available flight data could be FDR, voice, maintenance data, safety reports, cockpit display messages, or weather data. Analysts may wish to detect anomalies in flight operations, components, crew actions or even predict system failures.

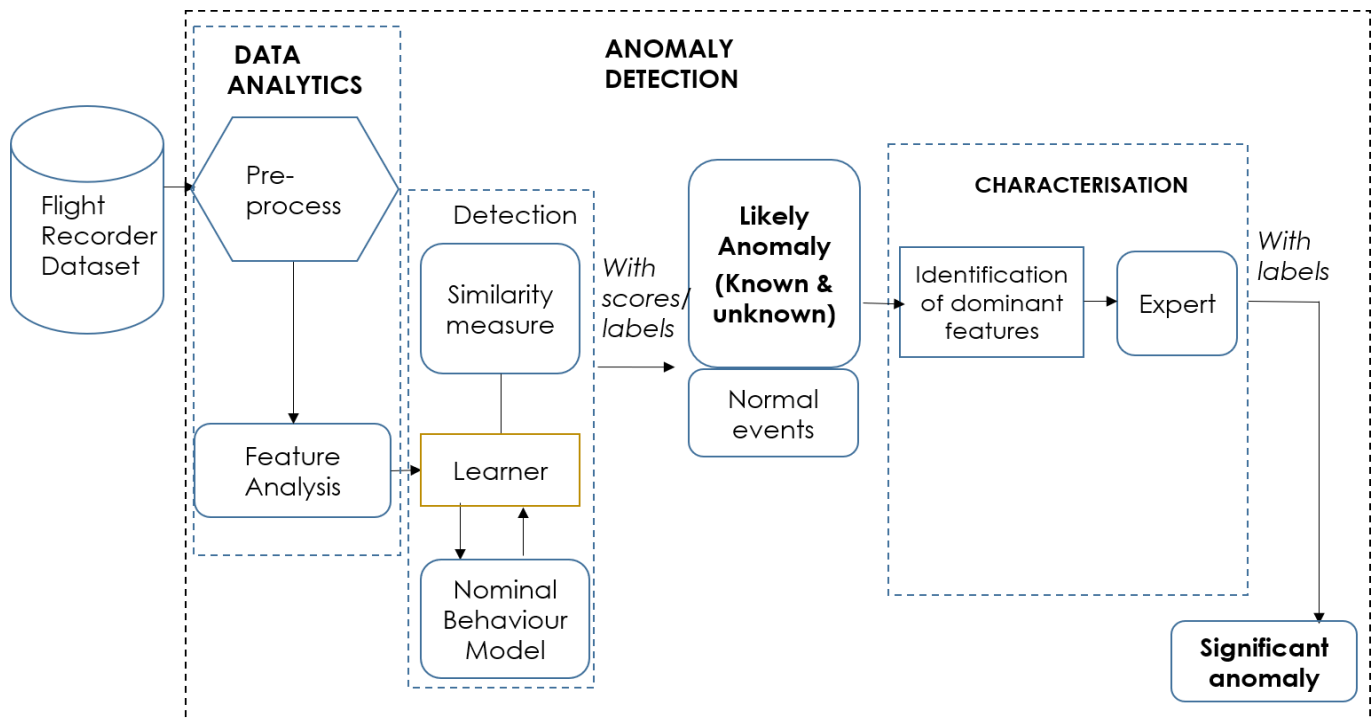


Figure 2-4 General AD Architecture

Figure 2-4 gives an overview of the generic AD architecture. Flight AD is generally based on the assumption that most of the flights in the dataset represent normal operations. The DA process thus involves preparing the data for analysis and the AD process begins with the output from the DA into a detection phase and then followed by the characterisation of the anomalies found as the output of the AD process. The detection phase generally entails performing similarity measures between data points thereby determining the nominal group and using this to ascertain the outlier groups; giving each output an anomaly score say, a zero-score referring to a nominal event. The characterisation phase involves identifying the dominant features associated with each anomaly then a domain expert validates its operational significance.

2.2.2 Anomaly Detection Applications and Techniques

AD has been proposed for use in different industries for diverse applications such as intrusion detection and fraud detection on networks/systems. In this work, the focus is only on the aviation sector and how various AD techniques have been applied to FDR data to detect unknown flight profiles and abnormal events. Applications of AD for flight data are condition-based monitoring, condition-based maintenance, and safety performance. Some of the current flight AD application areas and techniques are presented here and a summary of these application areas and respective techniques used is summarised in The **anomaly that an analyst wishes to detect** is also crucial to selecting an appropriate technique. Anomalies are categorized as Point, Contextual and Collective. Most AD techniques can detect an anomalous instance that is different from the majority set. However, not all techniques are able to detect data instances that are anomalies only in a specific context but not perceived as anomalies in other cases. This often occurs in time-series datasets like FDR data where it is possible to find scenarios where a pilot for example, carries out a sequence of actions in one flight phase, deemed as normal but the same sequence of events is deemed anomalous in another phase of the flight. Collective anomaly type refers to a collection of flights which are anomalous compared to other flights in the dataset. An example of collective anomaly type is where a few flights had similar anomalous throttle inputs compared to throttle input of most of the flight instances at the same period. In order to detect one or all these anomaly types, the appropriate AD technique will need to be applied. Finally, a flight technique selection will need to consider whether the goal is to produce an anomaly score (numeric value) or a label (normal or outlier) for every output flight instance. Since regression techniques require a class prediction of test instances, the output of these techniques will be labels categorizing the flight instance as a normal or anomalous flight. Hence, the need for the **Output selection criteria**.

2.3 Flight AD limitations and opportunities

The previous section has further highlighted that among existing flight AD techniques, none individually and satisfactorily handles the flight anomaly detection problem. In this section, a more robust data-driven hybrid AD opportunity for detecting a larger range of anomalies with a low false positive rate is proposed. Also, the challenges of SOP deviation detection and the opportunity of using AD to address them are reviewed.

Table 2-1 below.

Aircraft Condition-based Monitoring (ACM). This refers to the measuring of system parameters over time and while in operation. It is majorly used for predicting failure and calculating the remaining life trends of components. With ACM, replacement, for example, is only performed when the analysis of the data reveals degradation in the monitored components. Some AD methods have been proposed to help anticipate aircraft system failure risks as summarised in the next paragraphs.

The Aviation Performance Measuring System (APMS) was developed by NASA aimed at going one step further than ED by examining trends in multiple flights rather than single flight analysis [14]. Further to this was the Morning Report [15] developed in the late 90s as a software tool developed by NASA. The Morning Report relies on statistical means to identify atypical flights from multivariate comparison with baseline typical flights and tells which parameters contributed to an anomaly in a flight phase. This tool was designed for individual airlines attempting to address the problem of balancing between analysis for each flight and across multiple flights. It also clustered the resulting flights to help interpret the anomalous event. Morning Report, however, only focuses on continuous data.

Other clustering techniques and data analysis approaches were introduced such as Inductive Monitoring System software (IMS) [16]. It is a distance-based AD technique applying an unsupervised algorithm and using incremental clustering to build models of the expected operation of the system on a set of normal data. The model is then used to test new data and the distance score determines if it is classed as a nominal mode or anomalous mode. A limitation of this method is its vector space-based method which becomes computationally expensive as point-by-point analysis is performed. IMS does not consider temporal patterns in data hence sequential dependencies in the data are not incorporated. Also, the Euclidean distance used for measuring the similarity between discrete values is limited.

Developed by [17], Orca is a k-nearest neighbour based unsupervised AD algorithm. It has a nested loop structure to calculate pairwise distances between data points but

uses a simple pruning rule to keep the time complexity significantly less than the number of data points achieving near-linear performance with high dimensional data thereby making it appropriate for analysing large datasets. It can process both continuous and binary data format. A limitation of this method is that since each data point is scored independently, anomalies in the time domain cannot be detected [18]. iOrca [19] is a scalable version of Orca since the developers of iOrca introduced an indexing strategy and an early termination criterion to make it scalable to extremely large datasets during processing [18] but still suffers from the time domain limitation of Orca.

Aircraft Condition-Based Maintenance (CBM). AD for Maintenance Operations was motivated by the possibility that large flight datasets could be useful in understanding undesirable events like aircraft failures and need for scheduled maintenance. Maintenance activities ensure that the system does not fail, is functional and available for operation.

A new data analysis algorithm [20] for FDR data was proposed focusing on improving airline maintenance operations. This new approach categorizes the FDR time series data into continuous, discrete, and warning signals according to their parameter characteristics. A type of k-nearest neighbour approach is used to detect abnormal patterns in FDR coupled with the basic data analytics process involving appropriate techniques for Data Filtering, Data Transformation, Feature generation, Correlation Relaxation and Dimensionality Reduction. The performance of this algorithm was tested with NASA data. Results showed that it can detect unusual patterns within FDR regardless of parameter types thereby helping domain experts interpret data in less time for high-level fault diagnosis.

Safety Performance. This refers to the reduction to an acceptable level, risks associated with the operation of the aircraft and/or other aviation activities. Its continuous enhancement is deemed a fundamental objective of the ICAO and other aviation organisations. Many AD techniques have been developed to tackle the requirements for safety enhancements.

Sequence Miner technique [21] was developed as an unsupervised method to analyse discrete parameters and reveal atypical flight sequences. Only discrete parameters were considered, and it solves the problem of detecting and describing anomalies in large datasets of high-dimensional symbol sequences. It does this by using the nLCS (normalized Longest Common Subsequence) based distance measure. It can detect anomalies across a fleet of aircraft. It is, however, limited to analysing similar flights and aircraft since discrete sequences can vary greatly depending on the route, airport, and equipment. It is unable to integrate its findings of discrete actions with other continuous flight parameters since its performance degrades with interaction with continuous data.

One of the more recent techniques is the Multiple Kernel Anomaly Detection commonly known as MKAD [22]. It has been extended and investigated for small and medium-sized medical and finance datasets [23]. MKAD is a semi-supervised algorithm proposed for AD. Heterogenous parameters (continuous and discrete) were handled together. Discrete binary switch sequences are used in the discrete kernel and discretized continuous parameter features are used to form the continuous kernel. MKAD combines multiple kernels into a single optimization function using the One-Class Support Vector Machine (OCSVM) framework. The OCSVM uses a training set of nominal examples (in this case, flights) and evaluates test examples to determine whether they are anomalous or not. It combines the strength of both vector space and sequential AD techniques to allow a wider range of anomaly detection. Its ability to combine information from both data types makes it able to identify more complex behaviours of the crew in the anomaly detection process, unlike previous methods. It is, however, more able to find anomalies in discrete data than it can detect those from continuous data. For example, while it was able to detect anomalies in an auto-landing state due to an autopilot mode anomaly, it was unable to detect a high-speed approach based on relevant continuous parameters [24]. MKAD also needs nominal or near nominal data to learn proper hyperplanes, hence, is not appropriate for including multiple nominal data patterns since the trained model could identify other nominal patterns as anomalous.

ClusterAD [25] is a type of unsupervised learning method to detect an abnormal operation of an aircraft using FDR data. Euclidean distance was used as the similarity measure for both continuous and discrete parameters thereby giving rise to the possibility of incorrectly computed discrete parameters. A density-based clustering algorithm called Density-Based Spatial Clustering of Applications with Noise (DBSCAN) was used also to automatically determine the number of clusters, progressively find clusters based on density criterion and find outliers in the feature space, but an output of this algorithm is sensitive to its design parameters [26]. ClusterAD has also been used for flight AD through its combination with the LoOP – **Local Outlier Probability** machine learning technique [27]. ClusterAD is tuned to find anomalies in FOQA type data but does not clearly define the similarity measure of discrete data thereby producing a low performance in finding issues in discrete data [28]. It also loses temporal information during the reduction step thereby making interpretation of anomalies harder and even missing some vital short-length anomalies.

[29] contributed to improving aviation safety by proposing a general methodology of combining completely unsupervised learning techniques with supervised expert anomaly identification for large volumes of data. The method involves using diverse techniques for data pre-processing, data reduction, applying clustering algorithm, determining nominal and outlier cluster groups, identifying significant feature parameters associated with the anomalous group, domain expert characterization of special nodes and anomalous behaviours within the system. It further details how experts work with the anomalies produced and how they may extract information for decision-making. The work focused on the take-off phase of flight and was able to discover interesting anomalies such as a dead engine, near stall and a pilot choice to delay switching to autopilot mode soon after take-off. This work seeks to use a similar method developed by the author but proposes a hybrid AD technique to address the limitations of existing methods while combining their individual strengths.

The **anomaly that an analyst wishes to detect** is also crucial to selecting an appropriate technique. Anomalies are categorized as Point, Contextual and Collective. Most AD techniques can detect an anomalous instance that is different from the

majority set. However, not all techniques are able to detect data instances that are anomalies only in a specific context but not perceived as anomalies in other cases. This often occurs in time-series datasets like FDR data where it is possible to find scenarios where a pilot for example, carries out a sequence of actions in one flight phase, deemed as normal but the same sequence of events is deemed anomalous in another phase of the flight. Collective anomaly type refers to a collection of flights which are anomalous compared to other flights in the dataset. An example of collective anomaly type is where a few flights had similar anomalous throttle inputs compared to throttle input of most of the flight instances at the same period. In order to detect one or all these anomaly types, the appropriate AD technique will need to be applied. Finally, a flight technique selection will need to consider whether the goal is to produce an anomaly score (numeric value) or a label (normal or outlier) for every output flight instance. Since regression techniques require a class prediction of test instances, the output of these techniques will be labels categorizing the flight instance as a normal or anomalous flight. Hence, the need for the **Output selection criteria**.

2.4 Flight AD limitations and opportunities

The previous section has further highlighted that among existing flight AD techniques, none individually and satisfactorily handles the flight anomaly detection problem. In this section, a more robust data-driven hybrid AD opportunity for detecting a larger range of anomalies with a low false positive rate is proposed. Also, the challenges of SOP deviation detection and the opportunity of using AD to address them are reviewed.

Table 2-1 Summary of flight applications and AD techniques

Flight AD Technique	Flight Application	Approach/ Methods	Strengths	Limitations
<i>Exceedance Detection</i>	ACM	Rule-based method	Detects previously known anomalies	Misses the unknown anomalies
<i>Morning Report</i>	ACM	Semi-supervised / Statistical method	Anomalies in continuous data can be detected	Does not operate on discrete attributes
<i>Inductive Monitoring System (IMS)</i>	ACM	Unsupervised algorithm/ Distance based	Operates with mixed data types	Does not detect effectively, anomalies in discrete attributes. It is computationally expensive. Sequential dependencies are not considered.
<i>Orca</i>	ACM	Unsupervised approach	Performs well with high dimensional data. Processes mixed data types	Anomalies in the time domain cannot be detected.
<i>Sequence Miner</i>	Safety	Unsupervised / clustering using nLCS algorithms	Able to detect mode confusion since it can detect discrete data anomalies.	Unable to handle continuous data streams.
<i>MKAD</i>	Safety	Semi-supervised / OCSVM algorithm	Detects operationally significant unknown anomalies in practise. Processes mixed data types.	Kernel building step is time-consuming and memory intensive for mining very large data sets. No evidence it can detect flights with short-length deviation of parameters. Does not effectively detect anomalies in continuous data.
<i>ClusterAD</i>	Safety	Unsupervised/ Clustering using Euclidean distance	Detects unknown issues. Better recognise abnormal behaviours by allowing multiple SOP. Robust to noise in data. Processes mixed data types.	Does not effectively detect anomalies in discrete attributes. No evidence it can detect flights with short -length deviations. Temporal information is lost in the reduction phase making certain interpretation of anomalies difficult.

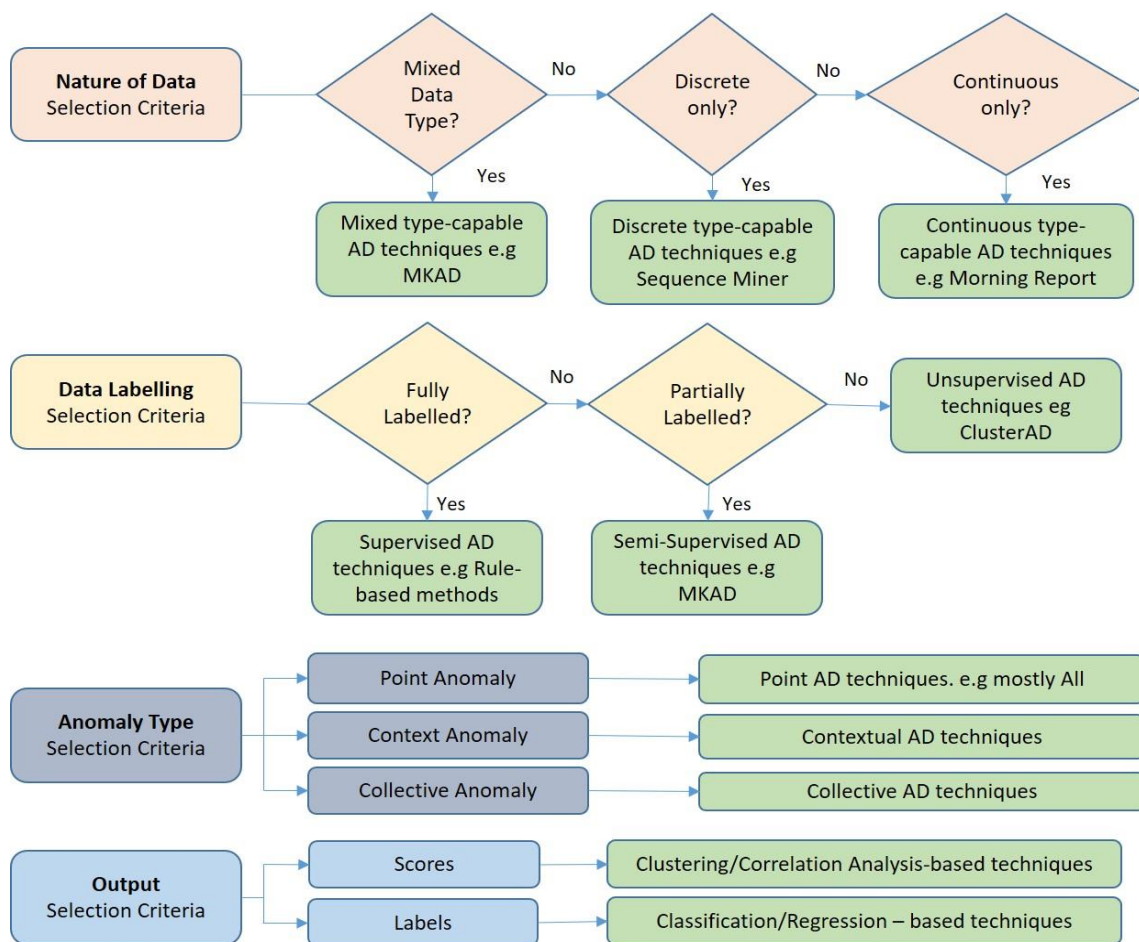


Figure 2-5 Overview of Flight AD Technique Selection Methodology

2.4.1 Hybrid AD Opportunity

The strengths and limitations of existing flight AD methods have been detailed in previous sections. Many AD techniques have been able to find component faults and operational faults but only a few of these [30, 31] are able to detect some type of anomalies of crew activity. Here, a hybrid technique to combine the strengths of multiple AD techniques with the aim of detecting a more varied range of anomalies than individual techniques is proposed. For example, while some techniques can detect anomalies influenced by deviations in discrete parameters, some are better at

effectively detecting anomalies of continuous variables. Moreover, some techniques are unable to identify nominal instances and some techniques that can, are only able to do so for certain portions of flight. Furthermore, while all the flight AD techniques mentioned were able to detect anomalies of discrete, continuous, sequences, long-length anomalies, there is no evidence that any would detect short-length anomalies. This could be due to the need for dimensionality reduction by these techniques which often filters out vital information [32]. Detecting short-length delays are important because [33] detailed how a 4.5s delay in engaging speed brakes at touchdown, caused the aircraft to overrun the runway leading to other incurred costs. A hybrid AD technique will be useful in combining the strengths of these multiple methods into one thereby finding anomalies in discrete, continuous, sequence parameters as well as inculcating the capability for detecting long and short-length abnormalities in the data.

Figure 2-6 gives an overview of this hybrid approach. The FDR data is pre-processed using appropriate methods including the selection of relevant features to ensure that vital information in the data is retained as much as possible. It is filtered for small portions of certain phases of flight to ensure all flights are comparable. The detection phase combines multiple data-driven techniques which output individually detected anomalies with their respective scores. The anomaly labels/scores can be numerical representing the extent at which a technique perceives the anomaly as significant. The hybrid AD technique includes a score merger function which takes the output scores of the multiple algorithms and outputs a final score that effectively represents how normal or abnormal an event is from the fleet-wide analysis. The scoring algorithm will use a detection threshold set at different sensitivity levels to ascertain the number of flights classed as anomalous thereby detecting, for example, the top y% anomalies in a set of flights. The score merger should also be able to make accurate judgements on anomaly classification in cases of conflicts in output between individual techniques. The final output of the hybrid algorithm is measured against baseline techniques and

validated using domain experts and/or available safety reports which in turn validate the actual significance of the anomalies found.

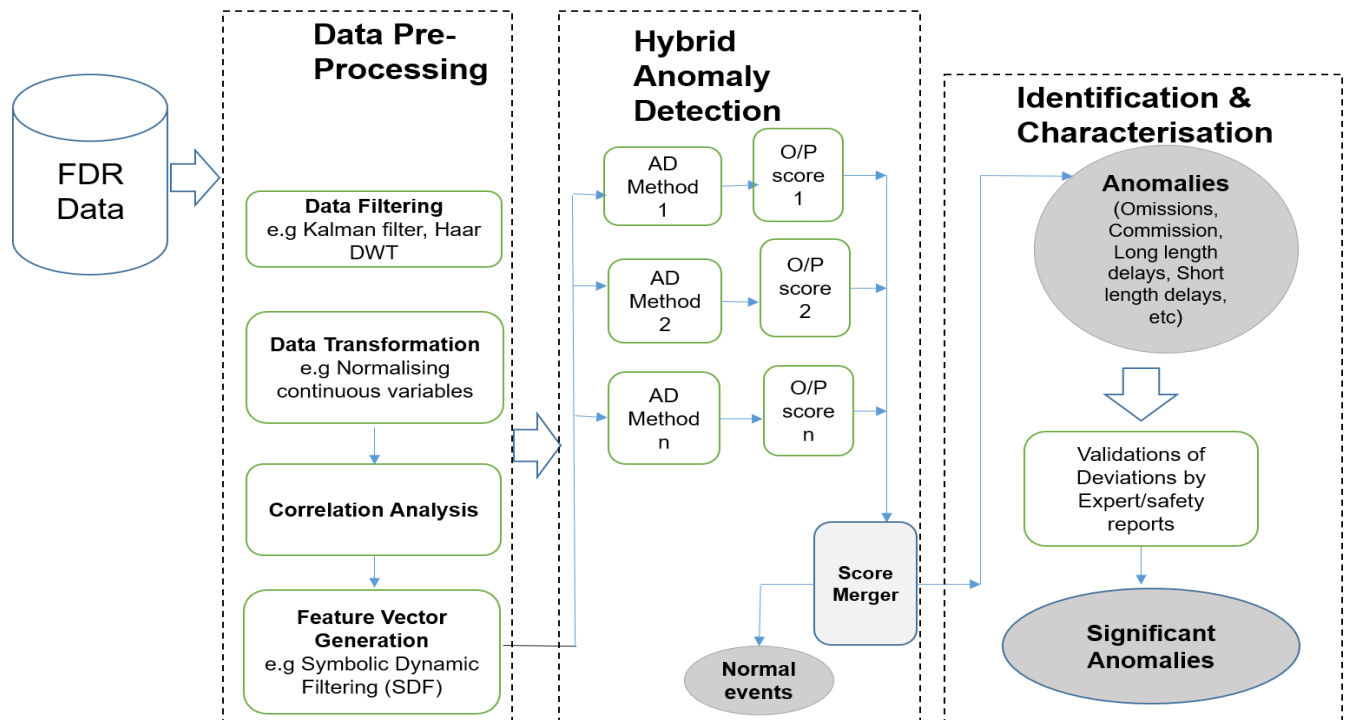


Figure 2-6 Overview of Hybrid Flight AD Approach

2.4.2 AD-SOP Opportunity

A Standard Operating Procedure or Checklist is a series of steps laid out to guide flight crew in appropriately and safely controlling the functioning of the aircraft.

Boeing reports that 80% of accidents today are caused by human errors [34]. In aviation, flight crew error could be in the form of pilot mistakes, oversights/omissions, commissions and failure to adhere to Standard Operating Procedures [35]. As commercial flights are highly standardized, aircraft operations are described through SOPs. Depending on the operational situation, the appropriate procedure is expected

to be applied by the crew. However, it is found that the deviations from SOP form a significant part of aviation accidents today involving loss of lives and other accident-related costs [36].

Possible scenarios that occur in the operation of a flight in relation to SOP adherence are cases where: 1) Procedures are followed but no accidents occur. 2) Procedures are not followed, and an accident occurs. 3) Procedures are followed but accidents still occur. 4) Procedures are not followed but no accident occurs. In order to proactively tackle the occurrence of SOP non-adherence, it is critical to focus on the latter scenario with a view to also detect incidents that occur due to SOP deviations during such flights so that mitigation strategies can be put in place to prevent any impending accidents.

Aircraft Condition-based Monitoring (ACM)	Aircraft Condition-Based Maintenance (CBM)	Safety Performance	SOP Deviation Detection using AD Opportunity
<ul style="list-style-type: none"> - Observes the state of the aircraft systems <ul style="list-style-type: none"> • Predicts component failures • Calculates component remaining life <p>Existing: APMS, Morning Report, IMS, Orca, iOrea</p>	<ul style="list-style-type: none"> - Carry out maintenance activities on the correct component at the right time. <ul style="list-style-type: none"> • Understanding undesirable events found during ACM • Scheduled maintenance planning • Increased asset life • Increased uptime & reduced downtime <p>Existing: DA technique proposed by Lee C.H et al</p>	<ul style="list-style-type: none"> - Reducing risks related to aircraft operation and human activity. <ul style="list-style-type: none"> • Less accidents and safety incidents • Detects known and unknown safety risks • Improvements to airport procedures • Proactive hazard identification • Identifies abnormal sequence of actions <p>Existing: Sequence Miner, MKAD, ClusterAD, Mixed method by Mack D.L et al</p>	<ul style="list-style-type: none"> - Reducing safety risks associated with flight crew deviations from procedures. <ul style="list-style-type: none"> • Identifies unknown deviations from procedure • Reveals the procedures used by crew • Shows extent to which crew actions adhere to SOP • Assess if procedure chosen by crew was appropriate • Better adapt training based on findings <p>Opportunity:</p> <ul style="list-style-type: none"> - Improve on existing knowledge-based crew deviation detection methods - Introduce unsupervised learning methods - Develop more robust hybrid AD technique for SOP deviation detection
ANOMALY DETECTION IN FLIGHT DATA			

Figure 2-7 Summary of flight AD applications and opportunity

There are, however, challenges in detecting SOP deviation. One of these is the difficult task of building a representation for every procedure (e.g., operator, airline and/or airport procedures) which requires a good knowledge of the domain. It is also challenging to analyse and isolate the source of every crew anomaly algorithmically. Furthermore, a case of normalization of deviance where an SOP deviation error among the majority of the crew over a long period has become perceived as normal, could occur.

The state-of-the-art outlier detection method is able to detect known procedure deviations but does not allow the discovery of emerging/unknown deviations from procedure. Some research [37-40] have addressed this gap by introducing expert-based methods for modelling SOP normal behaviour and then detecting unknown procedure deviation anomalies that arise. However, an improvement on modelling SOP with more unsupervised learning methods is desired.

There is, therefore, the need to explore flight data using Anomaly Detection framework to detect subtle underlying misunderstandings of the flight crew in relation to deviations from laid down procedures, which do not lead to accidents or incidents under most conditions or even hard to detect by the existing methods [41]. It is beneficial in not only implicitly considering the crew behaviour (in terms of a sequence of switches the crew may have triggered) as highlighted in Figure 7, but also the relation between pilot-related parameters and existing standard operating procedures. A data-driven AD approach can address the highlighted SOP deviation detection challenges by using patterns within the flight data to better identify and understand significant crew deviations that occur during portions of flights, simplify the validation work of experts by giving scores to anomalies, give more insight into understanding pilot aircraft operations and highlight ways to improve aircraft design or procedures.

The use of data-driven AD techniques in the detection of crew procedure deviations is a step in the direction of identifying leading indicators in safety management. This

could then serve as advance warnings of latent problems in training, procedure design or understanding automation to enhance safety. It also complements the state-of-the-art method by highlighting new exceedances related to crew activity by moving from a “was the speed brake deployed 1s after touchdown?” to “how many seconds after touchdown was the speed brake deployed?”.

2.5 Conclusion

As also summarised in Figure 2-7, this paper presented a review of flight anomaly detection techniques highlighting the application areas where they have been useful. The paper provides the strengths and limitations of these techniques as well as an emerging challenge of detecting deviations from standard operating procedures. It proposes the opportunities of using a data-driven hybrid AD method to address the limitations in existing techniques and the benefits of applying an AD approach to tackling SOP deviation challenges. The hybrid AD framework combines the strengths of multiple flight AD techniques to detect a wider range of anomalies than individual methods. A general overview of this approach is given from data pre-processing to the identification and characterisation of the detected anomalies by experts.

The primary contribution of this work is the discussion on the opportunities of AD approaches in improving existing limitations and emerging challenges for flight safety by carrying out a review of flight AD techniques. Specifically, it proposes a more robust data-driven AD method for the detection of a varied range of anomalies in high dimensional flight data by using a score merging function. The use of anomaly detection for aviation applications described in this paper provides avenues to complement existing detection methods and ultimately contribute to global safety enhancement initiatives.

References

1. Boeing, *Worldwide Operations| 1959 - 2017*, in *Statistical Summary of Commercial Jet Airplane Accidents*. 2017, Boeing Commercial Airplanes: Seattle, USA. p. 16-18.
2. FAA, *Safety Management System Manual*, T.O.o.C.S. Transportation, Editor. 2015, FAA.
3. Čokorilo, O., et al., *Costs of unsafety in aviation*. Technological and economic development of economy, 2010. **16**(2): p. 188-201.
4. Boeing. *Major Production Facilities*. Boeing Commercial Aeroplanes 3-05-2019 [cited 2019 5-05-2019]; Available from: <https://www.boeing.com/company/about-bca/>.
5. Zaslavsky, A., Perera, C., & Georgakopoulos, D. (2013). Sensing as a service and big data. *arXiv preprint arXiv:1301.0159*.
6. IATA, *IATA Forecasts Passenger Demand to Double Over 20 Years*. 2016, International Air Transport Association (IATA).
7. ICAO, *State of Global Aviation Safety*. 2013, ICAO: Montreal, Canada.
8. Cavka, I. and O. Cokorilo, *COST-BENEFIT ASSESSMENT OF AIRCRAFT SAFETY*. International Journal for Traffic & Transport Engineering, 2012. **2**(4).
9. Emmanuell Badea, V., Z. Alin, and R. Boncea, *Big Data in the Aerospace Industry*. Informatica Economica, 2018. **22**: p. 17-24.
10. ICAO, *FDA Limit*. 2016.

11. ATR, *Flight Data Monitoring on ATR Aircraft*. 2016, ATR Product Support & Services: Blagnac cedex - France. p. 16.
12. Biswas, G., et al. *An application of data driven anomaly identification to spacecraft telemetry data*. in *Annual Conference of the Prognostics and Health Management Society 2016*. 2016. Denver, Colorado.
13. Chandola, V., A. Banerjee, and V. Kumar, *Anomaly detection: A survey*. ACM computing surveys (CSUR), 2009. **41**(3): p. 15.
14. Statler, I.C., *The Aviation System Monitoring and Modeling (ASMM) Project: A Documentation of Its History and Accomplishments: 1999-2005*. 2007.
15. Chidester, T.R. *Understanding normal and atypical operations through analysis of flight data*. in *Proceedings of the 12th international symposium on aviation psychology*. 2003. Dayton, Ohio: Citeseer.
16. Iverson, D.L., *Inductive system health monitoring*. 2004.
17. Bay, S.D. and M. Schwabacher. *Mining distance-based outliers in near linear time with randomization and a simple pruning rule*. in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2003. ACM.
18. Das, S., et al. *Anomaly detection in flight recorder data: A dynamic data-driven approach*. in *American Control Conference (ACC)*. 2013. Washington, United States: IEEE.
19. Bhaduri, K., B.L. Matthews, and C.R. Giannella. *Algorithms for speeding up distance-based outlier detection*. in *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2011. ACM.

20. Lee, C.-H., et al. *Data analytics development of FDR (Flight Data Recorder) data for airline maintenance operations*. in *MFI 2017- International Conference on Multisensor Fusion and Integration for Intelligent Systems*. 2017. Daegu, Korea: IEEE.
21. Budalakoti, S., A.N. Srivastava, and M.E. Otey, *Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety*. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 2009. **39**(1): p. 101-113.
22. Das, S., et al. *Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study*. in *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. 2010. Washington DC, USA: Association for Computing Machinery.
23. Gautam, C., et al., *Localized Multiple Kernel learning for Anomaly Detection: One-class Classification*. *Knowledge-Based Systems*, 2019. **165**: p. 241-252.
24. Das, S., et al. *Comparison of algorithms for anomaly detection in flight recorder data of airline operations*. in *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. 2012.
25. Li, L., et al., *Analysis of flight data using clustering techniques for detecting abnormal operations*. *Journal of Aerospace information systems*, 2015. **12**(9): p. 587-598.
26. Ester, M., et al. *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*. 1996. Portland, Oregon.

27. Oehling, J. and D.J. Barry, *Using machine learning methods in airline flight data monitoring to generate new operational safety knowledge from existing data*. Safety science, 2019. **114**: p. 89-104.
28. Lee, S., I. Hwang, and K. Leiden. *Flight deck human-automation issue detection via intent inference*. in *Proceedings of the International Conference on Human-Computer Interaction in Aerospace*. 2014. ACM.
29. Mack, D.L., et al., *Combining expert knowledge and unsupervised learning techniques for anomaly detection in aircraft flight data*. at- Automatisierungstechnik, 2018. **66**(4): p. 291-307.
30. Shappell, S., et al., *Human error and commercial aviation accidents: an analysis using the human factors analysis and classification system*, in *Human Error in Aviation*. 2017, Routledge. p. 73-88.
31. Li, W.-C., *The Casual Factors of Aviation Accidents Related to Decision Errors in the Cockpit by System Approach*. J. Aeronaut., Astronaut. Aviat, 2011. **43**(3): p. 147-154.
32. Nanduri, A. and L. Sherry. *Anomaly detection in aircraft data using Recurrent Neural Networks (RNN)*. in *Integrated Communications Navigation and Surveillance (ICNS)*, 2016. 2016. IEEE.
33. Donald L. Wood, T.T.T., *The Effect of piloting Practices Upon Actual as Opposed to Scheduled Landing Field Performance*. AIAA, 2018(January).
34. Wiegmann, D.A., & Shappell, S.A. (2003). *A Human Error Approach to Aviation Accident Analysis: The Human Factors Analysis and Classification System* (1st ed.). Routledge. <https://doi.org/10.4324/9781315263878>.

35. Key Dismukes, L.L., *The Limits of Expertise: The Misunderstood Role of Pilot Error in Airline Accidents*. 2004.
36. CIAIAC, *Accident involving a McDonnell Douglas DC-9-82 (MD-82) aircraft, registration EC-HFP, operated by Spanair, at Madrid-Baraja Airport*. 2008, CIAIAC: Madrid, Spain.
37. Callantine, T.J. *The crew activity tracking system: Leveraging flight data for aiding, training and analysis*. in *20th DASC. 20th Digital Avionics Systems Conference (Cat. No. 01CH37219)*. 2001. IEEE.
38. Maille, N., *On the use of Flight Operating Procedures for the Analysis of FOQA data*, in *6TH EUROPEAN CONFERENCE FOR AERONAUTICS AND SPACE SCIENCES (EUCASS)*. 2015: Krakow, Poland.
39. Maille, N. *Modeling airline crew activity to improve flight safety analysis*. in *2017 IEEE Aerospace Conference*. 2017. Big Sky, MT, USA: IEEE.
40. Maille, N. *On the use of data-mining algorithms to improve FOQA tools for airlines*. in *Aerospace Conference, 2013 IEEE*. 2013. Big Sky, Montana: IEEE.
41. Jingru Yan, J.H. *Flight Data Monitoring and Human Factors Risks Identification: A Review of Best Practices*. 2013.

3 Unsupervised Labelling with Uncertainty Estimation as a Prerequisite for Flight Anomaly Detection

Flight Anomaly Detection (AD) methods are aimed at discovering abnormal events that occur in flight operations with a view to proactively identifying risks and mitigating them before undesirable outcomes are triggered. With the large amount of Flight Data Recorder (FDR) data that exists in the airline industry today, these AD solutions are now more data-driven and perform best when trained on labelled data. However, available datasets are almost universally unlabelled and in the airline industry (like other domains), data labelling is an expensive task since it is carried out by domain experts. In this work, an automated and cost-efficient approach to labelling flight data is proposed as a critical pre-requisite for improved AD solutions later incorporated in this research. The approach builds on previous work by integrating a user-in-the-loop, data visualisation and median-based Euclidean distance measures to label flight data as either anomalous or normal. In ideal circumstances, an anomaly will prompt further investigation by an expert however, in this paper, the focus is on determining the labels from the available partially labelled data. The AD can subsequently be carried out using supervised and/or semi-supervised methods with the labelled data from this phase as input. The approach is described and implemented on two flight datasets with some ground truth information from an expert and the results discussed. In addition, the proportion of uncertainty in the labelling technique is estimated using a probability method. The results show a 94% labelling accuracy, with an estimation of 12% uncertainty in the labelling approach used for the application studied in this paper.

3.1 Introduction

Spam filtering, customer segmentation, product recommendation, customer service automation, fraudulent transaction detection, image and voice recognition are some of the complex real-world problems to which Machine Learning (ML) is applied with the aim of finding effective and quick solutions. Anomaly Detection (AD) has been

used in some of these applications such as intrusion detection on networks for cyber-security, fault detection in safety-critical systems, system health monitoring, and event detection in sensor networks. AD is used in various domains and reveals data points with profiles that are different from the expected behaviour or characteristics [1]. In many application domains, the majority class represents the normal population. The minority class will be a mix of data points with erroneous sensor values and points with operational abnormalities both regarded as genuine outliers. The AD research area relies on a substantial amount of data to be able to build a robust model. Due to the vast amount of aircraft sensor data generated and collected in the airline industry, AD has also been proposed as an approach for detecting abnormal events in the health of the aircraft subsystems and in-flight operations as summarised in previous work[2].

3.1.1 Flight Anomaly Detection

Flight AD seeks to highlight anomalous system states that occur within the flight alongside their causal factors with a view to raising crew awareness, influencing operational procedures, or updating systems proactively to address unusual situations before they lead to unforeseen incidents in subsequent flights. The importance of Flight AD is revealed in the crash investigation report for flight AF447 in 2009. In this use case, the pitot tubes were blocked by ice thereby causing temporary inconsistencies with the airspeed information presented to the pilots on different displays. This meant that some automatic systems got disengaged, the pilots then needing to fly the plane manually at very high altitude thus leading to their inability to make a connection between the loss of airspeed information and the appropriate procedures for dealing with erroneous airspeed situation, the crew made incorrect inputs that destabilised the flight path and caused the accident [3]. Worthy of note here is that prior to this accident, nine other pilots had previously reported ‘unreliable airspeed’ occurrence in the airline’s Air Safety Report (ASR) stating that it led to temporary stall warnings. The aim of Flight AD is to detect anomalous events such as the confusing airspeed information, temporary stall warnings, etc that occurred in prior

flights for this use case, with a view to proactively tailor pilot trainings and/or make mechanical or policy changes to mitigate undesirable outcomes that could be triggered by these events.

Aircraft operations are highly standardized with established procedures for operating in every flight phase as well as procedures for certain abnormal occurrences that may occur. In such a standardized environment that helps the flight crew follow a safe routine, it is assumed that the majority profiles within any flight dataset would represent normal operations at that given time. In this study, “anomaly”, “outlier” and “abnormal” will be used interchangeably to describe flight profiles with unusual characteristics as depicted in available data. When these flight profiles are flagged, they are not necessarily unsafe until deemed as such by an expert.

The state-of-the-art approach used in the Airline industry to detect outlier events is called Exceedance Detection (ED) which flags up any anomaly that is greater than a set threshold in specific parameter markers. This method is typically designed to detect only known types of events. Current AD methods that have been proposed have been beneficial in finding unknown events and adequately characterising the causal factors. As an example, a study comparing Multiple Kernel Anomaly Detection (MKAD) and a Clustering AD (ClusterAD) method was carried out [4]. The authors contrasted the methods with the ED technique. A flight detected as anomalous by both AD techniques but classed as a normal flight by the ED method was identified as a ‘High Energy Approach’ meaning that the aircraft intercepted the glideslope from below, started to approach at a slower speed and a higher pitch than most flights which, ideally, should have warranted a go-around. The speed and power profiles for this use case were analysed and summarised to be likely precursors to runway excursion especially for shorter runways. This example shows the potential in flight AD which continues to be improved upon in literature.

The output of these flight AD techniques are the characteristics of the flights that caused them to be highlighted as outliers as well as anomaly scores which highlight the degree of abnormality for each anomalous flight in the dataset. To train flight AD models to reliably reveal abnormal events and attributes, accurate labelled examples are required as input into the models. Likewise, to evaluate the performance of these AD methods, their output must also be compared against known and accurate labels. However, with a lack of labelled data, the accuracy of AD techniques can become questionable.

3.1.2 Supervised Labelling Problems

Many domains use large data samples for solving ML problems with each instance requiring a label which is based on the values of its attributes existing within the data. For flight analysis, labelling generally entails accurately distinguishing flight instances or data points as either anomalous or normal flights. Ideally, this is a step that is carried out by domain experts using their knowledge of the relationships that exist between the variables of the flight hence, the supervised nature of the problem. The expert would analyse each flight and make a labelling decision based on the expert's knowledge of what normal looks like for the attributes and at specific times. The flexibility and understanding of real-world scenarios are what help experts make decisions based on common-sense and domain experience. However, with thousands of flights requiring labelling along with their multivariate nature, this fast becomes a daunting, expensive, and unrealistic task. The manual labelling can even be influenced by the annotator's personality and/or background. Therefore, the consistency in how labels are assigned to different samples may also be skewed when performed manually. The labelling problem is further compounded by the limited availability of time to label the data by experts. These reasons pose a supervised labelling problem which would benefit from an unsupervised labelling approach. This current work proposes an unsupervised approach to labelling huge data samples as a critical pre-requisite to more accurate training and evaluation of AD methods.

3.1.3 Unsupervised Labelling Solutions

Applying an unsupervised approach to this common supervised labelling problem is no trivial task. For safety critical domains like the aviation industry, there is no room for errors as they can lead to unexpected consequences. It is, therefore, essential to have accurate labels for building reliable ML models that, in turn, aid effective detection of anomalies in flight operations. An unsupervised approach to labelling data leads to more efficient use of scarce high-valued resource. It comprises of a model working unaided, to discover the patterns and information for an unlabelled dataset.

Although unsupervised learning approaches can classify data points, only the labelling decision of an expert is considered acceptable in specific domains [5]. Therefore, a user-in-the-loop [5] integration was introduced in this approach because of the need for higher accuracy in this very critical flight safety application. The proposed approach in this current work, first, makes use of the Euclidean distance metric to ascertain similarity measures between each flight instance and the median flight. A domain expert is then involved within the process to decide on an optimal threshold based on the similarity measures observed while visualising the sample distances and profiles. The threshold, in turn, adds to informing the appropriate label for each flight.

3.1.4 Uncertainty Estimation

There are some factors which make unsupervised methods prone to inaccuracies. Since the work presented in this paper proposes an ML labelling method, there is a need to consider the potential causes of such inaccuracies as well as determine a way of estimating the magnitude of the uncertainty in the proposed approach.

The presence of class imbalance is a natural property in the flight data domain that often reveals the normal class as having many more instances than the anomalous class. The existence of this kind of significant bias in the distribution brings to light the importance of assessing the uncertainty in the performance of learning models

implemented on the data. This is because a model is likely to incorrectly label a new instance as normal if heavily biased towards the majority class it has been exposed to in training have been normal samples. The second factor is uncertainty in assumptions made during the training process since. These assumptions are used in tuning the model hyperparameters which in turn influence how a model decides on the relationships between data points and then how the instances are labelled. The existence of edge cases within the data is another factor possibly responsible for inaccuracies that may exist within the unsupervised approach. Edge cases are flights with characteristics that position them on the borderline between the normal and abnormal class. Although these borderline flights are important in this safety critical domain, they are often ignored by many learning models because they so rarely occur.

The proposed Euclidean approach in the current work aims to reduce the uncertainty from biases, edge cases and assumptions. It does this by introducing user visualisation of the distribution of values to bring such potential error sources to light. The standard deviation metrics in the distribution, presented in the visualisation, is interpreted, and used to set the thresholds from which the labels are then determined. The labelling framework in this study proceeds to also estimate the level of uncertainty that exists in the resulting labelling; a detail that is often not acknowledged in literature.

The motivation of this current work lies in reducing the cost of and estimating the uncertainty in assigning labels to large unlabelled flight data, which is of high value for the subsequent flight AD process. This paper therefore makes the following contributions:

- To propose a median-based Euclidean distance metric in learning the similarities between flights to support the expert in the labelling process of a flight dataset
- Proposing a solution that also includes calculating the region of uncertainty in the labelling outcome

- A comparison between the proposed median-based distance approach and a density-based clustering technique using two datasets, to ascertain the best performing labelling method

The paper is structured as follows: section 3.1 is an introduction to the key concepts including AD, the labelling problem, unsupervised machine learning solutions and uncertainty estimation. This is followed, in section 3.2, by previous work carried out on key concepts and techniques used in this work. Then the proposed labelling process is described in section 3.3, and its results discussed. The uncertainty estimation approach is then explained alongside its results in section 3.4 and finally a conclusion on the approach is made in section 3.5.

3.2 Related Work

This section highlights the previous work on the approaches proffered for solving the labelling problem (Section 3.2.1) and existing methods for quantifying the level of uncertainty in the labelling results (Section 3.2.2).

3.2.1 Labelling

Traditional labelling entails querying domain experts about data samples and their corresponding annotations/labels are based on the expert's experiential knowledge. The labelling process is considered an important pre-requisite in developing effective machine learning models such as AD models. In other words, the labels become the ground truth information that aids learning in autonomous supervised models and the greater the number of labelled instances available, the better the learning. While some labelling such as spam flagging or movie rating require little cost, other more advanced labelling like speech recognition or flight anomaly detection are time-consuming, difficult and/or expensive to perform. More unsupervised techniques have been

proffered to reduce the costs involved in the classification process (or labelling process, in this case).

The workings of such unsupervised approaches to labelling assume that a group of points in the data will be assigned the same labels if they share a common relationship that is based on the properties of the data points considered as a whole. Depending on the relationships that are found among these data points, there could either be a single class or multiple classes (including binary classification as is the focus in the current study) but each data instance is expected to have just one label. For effective labelling, quality data is essential. In real-life problems, available datasets are either completely unlabelled or partially labelled. Learning relationships between datapoints with no labelled instances at all is referred to as the cold-start problem and is particularly challenging for model-driven strategies. In this current work, the dataset used is mainly unlabelled with only a handful of data points pre-labelled by an expert. The following sub sections highlight related work and describe the key concepts which were useful in developing the proposed labelling approach presented in this paper.

3.2.2 Active Learning (AL)

Active learning (AL) is an area of machine learning which, though similar in concept, is not to be confused with the instructional method with the same name in the education domain [6]. AL thrives in ML problems with a significant amount of unlabelled data. This field of AL is based on the hypothesis that a machine learning algorithm can be more accurate with fewer labelled training instances. The idea is that if the active learner chooses the data it learns from, it performs better but with less training data as opposed to the classical learning which requires hundreds or thousands of labelled instances to perform well [7]. This, therefore, results in a reduction in the cost of acquiring labelled information.

Systems that make use of the AL concept aim to tackle the labelling problem by requesting unlabelled instances to be labelled by an oracle (a human annotator). The

active learner then aims to achieve high accuracy using as few labelled instances as possible and making the choice of the most informative instances by using different strategies. One of such strategies is the uncertainty sampling query strategy which selects the instance in a pool of unlabelled instances about which the model is least certain how to label, that is, the instances close to the decision boundary [8]. Traditional AL approaches have been proven to perform better than strategies that randomly select candidate instances for machine learning tasks.

In the proposed approach, a human is similarly introduced into the training process as well as the use of a few pre-labelled instances as was carried out in the AL approach. However, in the current work, the pre-labelled instances are selected at random by the expert prior to learning but unlike in AL the pre-labelled set are not part of the training data. Instead, the pre-labelled instances are useful for ascertaining the performance of the proposed approach (i.e. comparing their predicted labels from the approach against their true labels from the expert). The benefit of having a user alongside learning models in the labelling framework to aid a higher accuracy for the application and the reduce costs compared to manual labelling, has been re-iterated [9]. In a two-class labelling problem, which is the focus in this study, it was also shown that the classic active learning can be outperformed by Visual-Interactive Labelling (VIL) [7].

3.2.3 Visual-Interactive Labelling

The VIL approach [5] entails the incorporation of a user to interactively label the data points using visual exploratory aids. Incorporating visualisation into the labelling process is considered necessary as it aids discovery of insights into complex data by the user. This method is quite popular in the visual analytics domain as it allows users to acquire useful information from the data. The information gathered is beneficial for increasing their knowledge and ultimately help in the labelling decision-making. The information from the data could be in the form of scores, rules or similarity measures and this variation in the type of information garnered, constitutes the basis for the

difference in VIL approaches [5, 10] in literature. The work presented in this study seeks to perform better in the annotation of the unlabelled samples by also adopting the visualisation method for the user. A distance measure is introduced while integrating it with visual aids that provide useful similarity information to help the user define the decision boundary which in turn, determine the data labels.

3.2.4 Euclidean Distance Similarity Measure

The Euclidean distance provides a measure of absolute distance between two points. Its general form, for points p and q in any dimensional space, is given in (3-1)

$$dist(p, q) = \sqrt{\sum_{i=1}^n (q_i - p_i)^2} \quad (3-1)$$

where i represents the i^{th} dimension. Classical distance-based models tend to use the centroid/mean of the raw data to infer decision boundaries however when the distribution of the data is heavily skewed, the centroid property becomes misleading in identifying the relationships between the instances and in labelling them. To explain this, consider a population of 10 numbers with two of them having much higher values than the others. The data becomes skewed, and the mean of the data is not representative of the distribution. In this case, the median would give a better representation of the data.

Consequently, the median-based method was chosen to replace the traditional centroid/mean-based approach. The median is a popular metric for descriptive statistics and is a suitable measure known, generally, to be robust, fast, and comparable in performance to more complex methods [11]. As the domain context moves from one-dimension to higher dimensional data applications, multi-variate medians can become complex in their computation. This is because there is no natural way of ordering data points in multiple dimensions unlike in univariate spaces where you simply order the instances and select the middle data point. This problem has given rise to several methods [12] each aimed at defining some ordering function in

the multivariate space before computing the median thereby making it less interpretable. Parameter-wise medians are hence, preferred and used in the current work. The median functionality allows for building of a better model that is less sensitive to extreme outliers. This study also makes use of parameter-wise medians to build the high dimensional data Euclidean model. It is expected that this technique, when merged with VIL for a user-in-the-loop, will yield improved accuracy when applied on flight datasets.

Traditionally, a median-based distance technique would require a median for each unique class population. For a binary classification problem, the classical median approach uses two parameter-wise medians to determine if a data point belongs more to one class than the other class (in this case, the anomaly class or to the normal class populations). The formulae in equations (2) and (3) show how this decision can be reached in theory. Specifically, $ALabel_p$ and $NLabel_p$ represent the anomalous and normal labels that can be assigned to a random data point, p respectively. $AMed_i$ and $NMed_i$ are the component-wise medians of the abnormal and normal class populations respectively. For example, if $|AMed_i - p_i|$ is 10 and $|NMed_i - p_i|$ is 20 then the difference of -10 means the random point, p would be assigned a normal label, $NLabel_p$ since it is less than 0.

$$ALabel_p \leftrightarrow \sum_{i=1}^n |AMed_i - p_i| - |NMed_i - p_i| > 0 \quad (3-2)$$

$$NLabel_p \leftrightarrow \sum_{i=1}^n |AMed_i - p_i| - |NMed_i - p_i| \leq 0 \quad (3-3)$$

Therefore, the data point, p of the i th component would be classed as anomalous if and only if the absolute distance from the median point of the normal population, $|NMed_i - p_i|$, is subtracted from the distance from the median point of the anomalous class, $|AMed_i - p_i|$ and the result is greater than 0 [11] as represented in (3-2). Conversely, if the distance measure resulted in a value less than or equal to zero as expressed in (3-3), then a normal label is assigned to the data point, p .

However, due to the hugely unlabelled nature of the data in this current study, using two median profiles is not possible as class membership is not known a-priori. Therefore, one median profile for each component is used and the label decision-making process that occurs afterwards will be explained in Section 3.3. Imbalanced data or class biases in the data coupled with errors that may exist in the assumptions made by the model and user, give rise to the need for measuring the level of uncertainty in the labelling approach.

3.2.5 Uncertainty Estimation

Uncertainty estimation, in this study, measures how unsure a model is at labelling flights in the dataset. Although there have been major strides in the performance of machine learning models over the years, factors such as biases/imbalance in the data, presence of edge data instances and high dimensional nature of data could give rise to incorrect representation of reality and in turn inaccurate labelling. For domain stakeholders, it is not sufficient to simply acknowledge that models can be unsure/uncertain about their output, but the uncertainty should be appropriately estimated to inform their decision to embrace a specific labelling outcome.

Previous work [13] corroborates the need to estimate the total model uncertainty from all factors/sources as opposed to measuring the uncertainty only in individual sources. The authors focused on the water resources management domain and applied an uncertainty estimation method that is based on analysing the model errors using an extended clustering-ML technique. Like in many other uncertainty calculation techniques, the model error (which is only one of the many other contributing factors to uncertainty) in the extended clustering-ML approach is borne from the discrepancy observed between the target and the predicted information. Therefore, the availability of both target and predicted label information, aids the uncertainty assessment of the method. This approach would, however, be infeasible for a scenario like the current

flight AD study which is without complete target labels against which the model prediction can be compared.

Another study [14] introduces an Active Learning strategy that combines model-based recommendation of instances with a user-VIL for labelling instances until the system has enough candidates for training the model. The authors calculate the uncertainty in the candidate classification that the model makes to the user. For each instance, the model learns the probability of the two classes resulting in a vector of these probabilities which is presented to the user. The uncertainty in selecting an instance with high benefit for training, rests on the margin between the probabilities of the two classes. The complement of the margin gives the estimated uncertainty of the model in suggesting a useful candidate as represented in equation (4). In other words, ‘complement’ denotes that the margin and the uncertainty are mutually exclusive while adding up to 1 as exemplified in a *low margin* between the class probabilities being equivalent to a *high uncertainty* in the class assigned to that instance. Since the model ultimately aims to clearly distinguish between classes (i.e., produce a lower label uncertainty), the authors therefore introduced a margin approach to increase the margin between these class probabilities so that there is a reduction in the error produced by the model’s classification.

$$Uncertainty(x_i) := f_{uncertainty}(x_i) = 1 - f_{margin}(x_i) \quad (3-4)$$

To explain the margin approach further, a probability range between 0 and 1 with 1 being the possibility of an instance being anomalous, is used. If the probability of an instance x_i being anomalous is 0.6 and the probability of the same instance being normal is 0.4. This gives a small probability margin of 0.2 highlighting that the model is very unsure or uncertain of the label for that sample. Using the uncertainty calculation in (3-4), the model’s uncertainty for this instance is $1 - 0.2$ which gives 0.8 or a high uncertainty in the recommendation for that instance.

The data points that fall around the decision boundary of a model often have the highest uncertainty since it is here that the highest degree of confusion exists for the model. Similar to [14], this current study also employs the use of probabilistic methods to define the uncertain classes around the boundary thereby forming a region of uncertainty that can then be estimated. The proposed uncertainty calculation approach is explained in more detail in section 3.4 of this paper.

3.3 The Labelling Framework

The process of labelling flight data instances is demonstrated and evaluated using two datasets – a publicly available National Aeronautics and Space Administration (NASA) synthetic flight dataset [15] and an operational flight dataset from an international airline that is not named for confidentiality reasons. Both datasets are time series sensor data from the Flight Data Recorders (FDR) of a fleet of aircraft. They are also similar in having multiple parameter information (e.g., altitude, airspeed, flap positions, etc) and heterogeneous datatypes (i.e., continuous, and categorical). A small set of instances from both dataset pre-labelled by a domain expert are used to tune the performance of each method.

Figure 3-1 below describes the labelling workflow which begins with the pre-processing of the raw FDR data involving steps such as dealing with missing values, feature selection, data transformation and filtering. The next step involves the selection of the best labelling method between two clustering methods as well as the proposed median-based approach. This phase is built on the premise that any of these methods that performs best in labelling the NASA dataset, will also be best fit for labelling the larger operational dataset since both datasets are similar in structure and content. A validation step carried out by the domain expert on representative instances, confirms the overall best labelling method of the three and this chosen method becomes the labelling approach proposed and described in detail in this paper. The flight labels for the entire dataset produced by this approach becomes the ground truth information

used in the subsequent AD task for training and evaluation. An extension to this labelling solution is the uncertainty estimation of the selected approach which is the last phase of this framework. The following sub sections explain these steps in more detail.

3.3.1 Data and Pre-Processing

Before any analysis can occur, a significant part of the data analytics process focuses on extraction, pre-processing, feature engineering tasks and generally preparing the data for analysis.

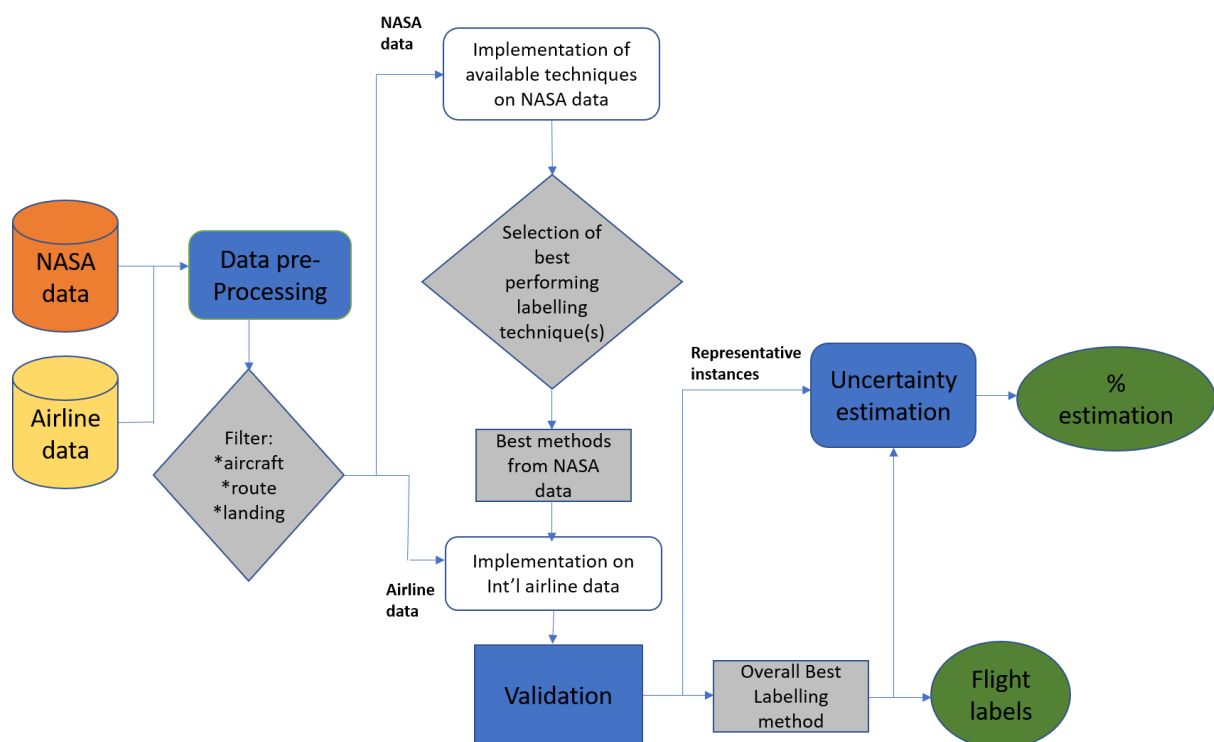


Figure 3-1 Building blocks of the Labelling Framework

3.3.1.1 The Flight Datasets

The inputs into the framework are the two timeseries datasets – the operational data from the international airline and the synthesised data from NASA made public for testing their MKAD algorithm. FDRs measure and record operational information from aircraft systems. Previous work [2] summarises how FDRs operate and how the data is extracted from the sensors and aircraft.

For this study, the operational dataset used is from an airline that has now ceased operations and it focuses on the FDR from their Airbus A320 aircraft fleet. To ensure the flights are comparable to tackle the subsequent AD problem, flights on the same route (i.e., from one specific departure airport to a specific destination airport) are chosen but covering a 10-year period. However, it is worthy of note that the dataset on this route consists of multiple runways at both the departure and destination airports. Therefore, the data is fleet and route-specific but not runway-specific. An average flight on this route takes approximately 1.5hours and the number of flights available for analysis totals 10,152. Each raw flight data was extracted with 370 parameters, sampled between 1 and 8Hz, consisting of some categorical parameters such as flap position, thrust modes, auto pilot modes and continuous parameters such as airspeed, groundspeed, altitude, roll angle and pitch angle. The flight files are extracted and converted into individual comma separated value (csv) files, one for each flight. Table 3-1 below shows an example extract of raw data for one flight. This dataset was largely unlabelled except for 16 randomly selected flights (7 anomalous and 9 normal) that are pre-labelled by a domain expert. The pre-labelled flights are used as the initial ground truth information for this airline dataset and used for tuning the hyperparameters of the labelling techniques implemented in this current work.

Table 3-1 Section of raw flight data example from the operational airline dataset

Time_(secs)	Computed_Airspeed	Ground_Speed	Roll_Angle	Thrust Mode	Flap_Angle
4592	130.9	128	-2.81	*NOTACT (0)	40
4591	133.9	128	-1.05	*NOTACT (0)	40
4590	135.5	128	-2.11	*NOTACT (0)	40
4589	137	128	-0.35	Active (1)	40

The publicly available NASA dataset[15], on the other hand, consists of 112 flights. In its original form, it has been prepared with a consistent 1Hz sampling rate and typically consists of a 1.5-hour flight duration. Table 3-2 shows a few of the 13 parameters for this dataset with its heterogeneous (categorical and continuous) variables and each flight is also available as a csv file. This dataset was necessary for this study for selecting the labelling method. The selection process on the NASA dataset, therefore, also serves as the first step in validating the proposed approach. The initial ground truth information available for this dataset are 33 flights known to be normal. Consequently, the labels of the remaining 79 flights are unknown.

Table 3-2 Section of flight example from the synthetic NASA dataset

Time	AirSpeed	Altitude	Landing_Gear	Param2	Param4	Reversers
5443	82.224	9956	0	-101.75	44.215	0
5444	82.864	9952.7	0	-101.97	44.225	0
5445	83.527	9928.8	0	-102.94	44.36	0
5446	84.206	9911	0	-102.62	43.766	0

3.3.1.2 Data Preparation

Unlike the opensource NASA dataset, the airline dataset is provided in its raw form and requires cleaning and transformation prior to it being used for modelling. In this part of the labelling framework, feature selection based on domain expert knowledge was carried out giving rise to 48 features (16 categorical and 32 continuous) selected for the remainder of the process. The data preparation process, carried out in Python, also included dealing with missing values, stripping of unwanted characters, encoding categorical variables, and renaming parameters. The pre-processing also involved transforming the data to a 1Hz sampling rate thereby extracting the operations for every second of the flight and then converting to parameter-wise vectors. The data filtering entailed removing erroneous flight data (emanating from sensor malfunction revealed as incorrect data values) that could introduce noise to the process or skew the machine learning process in the long run.

The landing phase was selected as the specific area of focus for this study because it is known to be a part of the flight with a high incident rate. In this work, incidents in the landing phase refer to situations that tend to arise in landing that do not happen in other parts of the flight such as go-arounds, improper landing configuration, etc. Using the touchdown point for each flight as a point of reference in extracting the landing window was useful as it prevents the flight samples from having variable lengths thereby ensuring the landing window of all flights are comparable. In preparing the data, the 1.5-hour duration of the flight was reduced to a 90-second window focusing on the final part of the landing phase. The 90-second window spans 70 seconds prior to and 20 seconds after touchdown (defined as the point at which wheels make initial contact with the ground). This short window within the landing phase was selected to capture the increased pilot activity expected from final approach to flare, landing roll, touchdown, and part of rollout.

3.3.2 Labelling Method Selection using NASA Dataset

The aim of this section is to consider unsupervised methods and select the most robust and accurate method for labelling flight data while implemented on the NASA dataset. These techniques are first applied to the synthetic dataset and their performance evaluated against the 33 known flight labels. Carrying out this process, initially on the synthetic NASA dataset, acts as a first validation step for the technique eventually implemented on the operational airline dataset especially since both datasets are similar in structure and both cover the landing phase of flight.

In this section, a few methods are introduced with a description of how they work, why they were chosen for this labelling task, their implementation on the NASA data and their labelling performance. These methods are Density-Based Spatial Clustering Algorithm with Noise (DBSCAN), K-Prototype and the proposed median-based method with VIL.

3.3.2.1 DBSCAN

DBSCAN [16] is a popular clustering algorithm that has been implemented in python toolkits such as scikit-learn. Its model assumes that clusters consist of dense areas separated by regions of lower density. Specifically, a hyper-circle is created around a data point (*Core Point*) to outline the required density as well as the radius of this circle referred to as *Epsilon*. Within this circle, a minimum number of similar points (*minPoints*) are considered as belonging to the same cluster forming a separate nominal group. Data points within the same cluster but are not core points are referred to as *border points*. This process is carried out for every data point and where any of these border points becomes a core point, its neighbours also become *density connected* [17] into the same *core set* created. Data points outside of this set that are not reachable from any core point are called *outliers/noise* and form the lower density areas separated from the core set.

This clustering method was selected for the current work because it possesses the advantage of being able to optimally group data points in an irregularly shaped dataset like the flight data. It is also beneficial in highlighting outliers and can automatically define the appropriate number of clusters without being fed the information in advance as is the case with some other clustering algorithms. The only hyperparameter values that are set (based on the attributes of the specific data) are *Epsilon* and *minPoints*. Both values affect how the model is trained as well as the efficacy of its classification. For each core set, *minPoints* defines the number of flights that make up those normal cluster groups and *Epsilon* influences the proportion of anomalous flights that can be found [4]. Both values are determined with multiple runs of the algorithm while noting how it classifies the pre-labelled instances. Pairwise distances between points are used to decide on the appropriate epsilon since the runs are carried out using a range from the minimum to the maximum pairwise distance in the data distribution. The optimal *minPoints* value was determined from a heuristic method called *Elbow method* where the elbow of the curve gives the required value based on the premise that an increase in the value does not produce any better modelling of the data. In this use case, an *Epsilon* between 5-10 resulted in the same abnormal points being revealed and so, a fixed value of 5 was preferred and used.

Since the landing phase is highly standardized, it is expected that majority of flights would cluster together giving rise to the higher density population being associated with the normal label class that is, the most common operations for landing. Other emerging lower density clusters reveal less prevalent landing profiles that could either be a different type of normal or variations of abnormal [18]. It was expected that the pre-labelled instances would fall into similar clusters that represent the appropriate class label. The clusters in which the known instances fall, gives the user a good insight into the characteristics of the other flights that make up the clusters. Consequently, entire clusters with constituent flight data points can be assigned the same label as those assigned to any of the 33 known instances that fall within them. Where emerging clusters do not include any of the 33 instances and are well separated

from the core cluster, such clusters and their data points are treated as anomalies. The pre-labelled instances act as initial ground truth information that help produce a higher accuracy in the performance of the methods used in this work. The performance details are discussed in section 3.3.3.

3.3.2.2 K-prototype

K-prototype [19] is another technique that was considered and implemented on the NASA dataset to assess its performance in labelling flight data. It is an extension to two popular algorithms: K-Means and K-Mode. K-Prototype was developed to address the challenge that arises from modelling in a heterogeneous/mixed data environment. In this case, the challenge occurs because most machine learning algorithms require numeric input quantities. K-Means for example, though efficient for large datasets, is unable to cluster categorical variables since its dissimilarity measure is calculated using the Euclidean distance with the mean metric for forming clusters. The K-Mode algorithm, on the other hand, is only designed for categorical data as it makes use of the mode metric for its cluster formation. Also, there is the challenge of contextual insights that exist within categorical variables that could be lost when an inappropriate distance metric is used. Therefore, the presence of mixed types in the flight data requires the use of an approach that can tackle both numeric and categorical quantities.

K-Prototype takes as input the heterogeneous dataset but with insight about the parameters that are categorical, and the number of clusters desired. K-prototype combines the K-mode functionality which is used to deal with these categorical features and the K-means clustering capability which handles the continuous data quantities. The integration calculates the dissimilarity between mixed type data points by combining the squared Euclidean distance measure on the numeric quantities with a matching similarity measure on the categorical attributes[20] and includes a weight which is adjusted based on the data type that dominates in the dataset. It then

produces an index of each cluster and indirectly the labels of each of the similar flights in the clusters. It was selected for this study because of the heterogeneous nature of the flight dataset and the capability of K-Prototype to deal with each constituent data type. Although the performance of K-Prototype was slightly better than its individual K-Means and K-Mode components, it was however not as accurate as DBSCAN in appropriately predicting a good number of the 33 known instances of the NASA flight dataset hence was not applied to the operational dataset.

3.3.2.3 Proposed Euclidean Distance Method

In this section, a median-based distance method combined with VIL approach is described for the labelling of flights in this safety critical domain. In the traditional median-based technique, two median profiles (one for each class) are required however, due to the unlabelled dataset, one median profile is used.

Figure 3-2 below summarises the steps involved in implementing the proposed method. First, a set of representative parameters (both categorical and continuous), that have the highest effect on the flight labels, are selected as part of the pre-processing phase.

For high-dimensional data, the feature selection step is common in practise as certain parameters may make little or no contribution to the labelling process. The benefits range from improved labelling performance, reduced processing time to understanding of influential parameters. After the parameter selection step, the timeseries data from the different flights are mapped into comparable vectors in the high dimensional space.

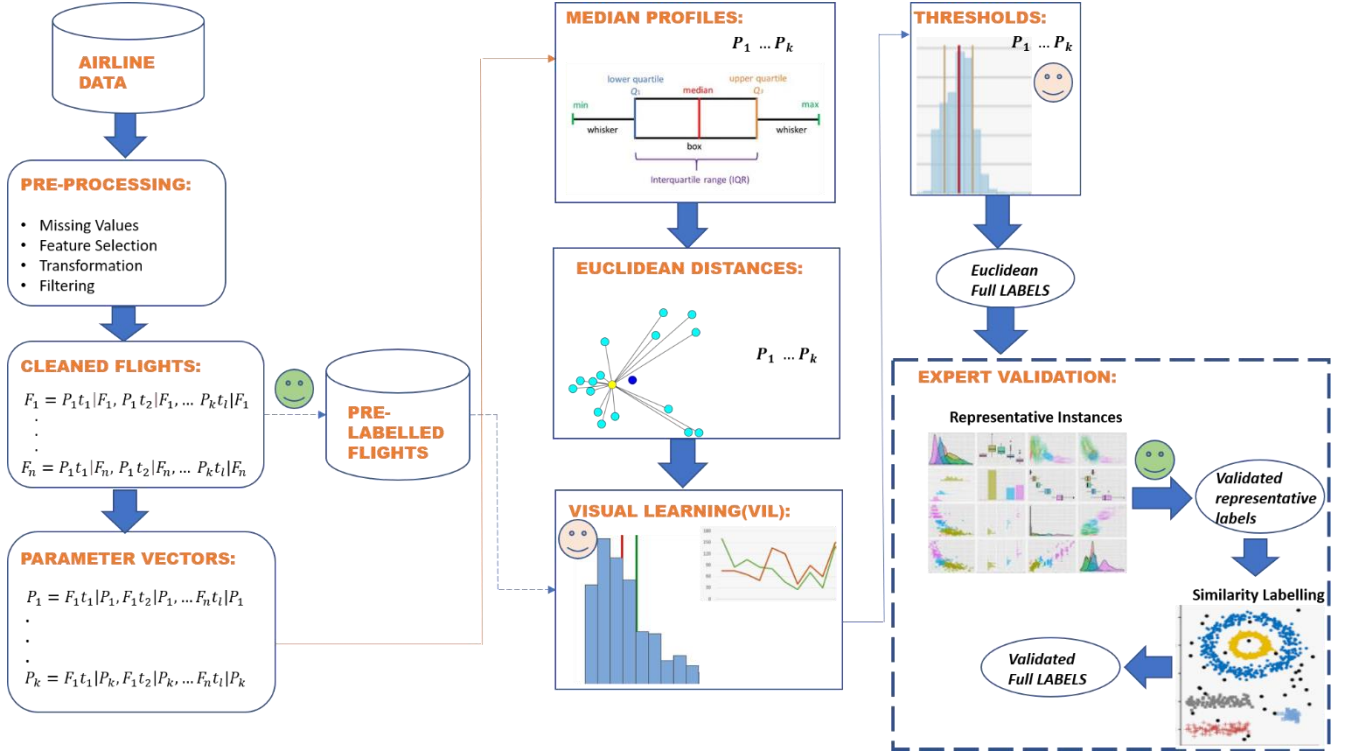


Figure 3-2 The framework for proposed median-based distance with VIL approach

Equation (3-5) represents the sampled values arranged to form a vector comprising all flight values for each parameter. $F_j t_m P_k$ is the value of the k th parameter at sample time t_m for flight F_j ; n is the total number of flights, i is the total number of time samples for every flight and for the specific parameter in focus. An example is shown below in Table 3-3 providing information for say, the airspeed, of all flights and at each of the 90 seconds time samples of the landing window. This is carried out for each parameter, that is, P_1 to P_k .

$$[F_1 t_1 P_k, F_1 t_2 P_k \dots F_n t_1 P_k, F_n t_2 P_k \dots F_n t_i P_k] \quad (3-5)$$

Table 3-3 Sample of transformed parameter-wise data for the airspeed parameter

	t1	t2	...	t89	t90
Flight 1	137.4	138.9		63.8	61
Flight 2	130.4	129.4		46.5	44.6
...					
Flight n	159	159.1		55.5	52.8

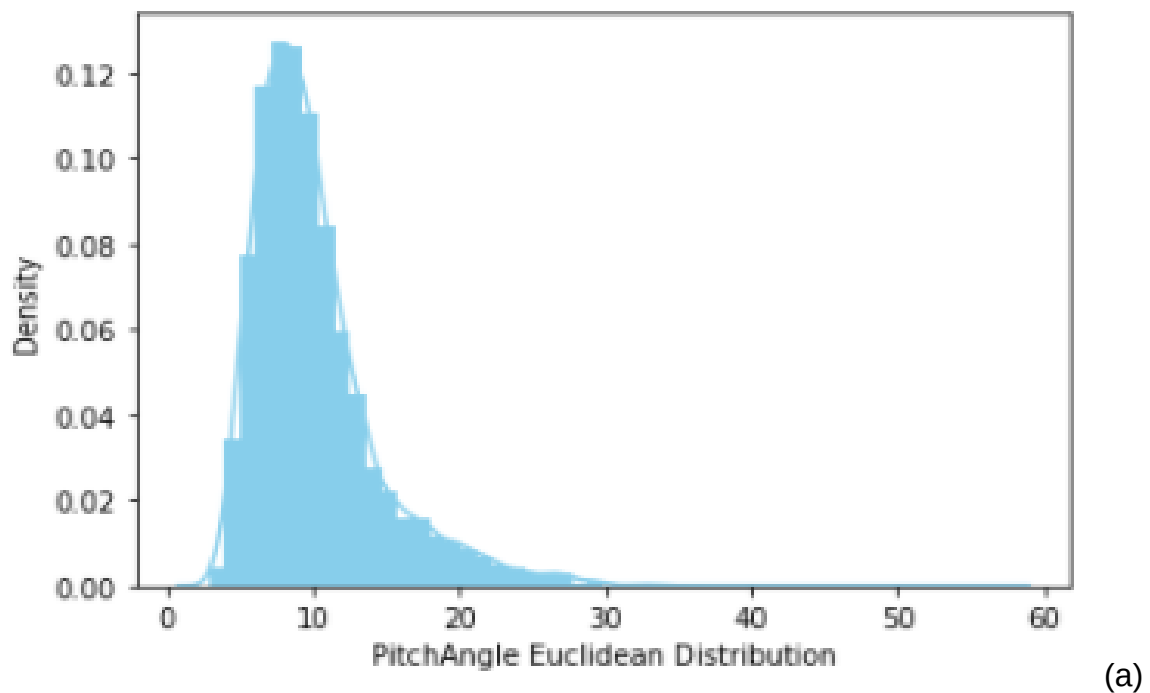
The output of the pre-processing phase is a clean version of the timeseries flight data with reduced parameters and in a suitable vector space ready for analysis. From this univariate feature space, the median profile of the landing window is determined from the list of timeseries flight profiles. A list of distances highlighting the distances from each data point to the median flight profile is then calculated.

Inspired by the VIL approach, this method presents the distances on visual aids such as a histogram like in Figure 3-3 below. With the visualisations and the prior knowledge of some pre-labelled instances, a user can gain insights on the different flight operations revealed in the data, make informed assumptions about best distance threshold for each univariate vector space and ultimately make appropriate labelling decisions. To validate the labels produced by this method, representative flight instances from each operational group are extracted and presented to an expert to confirm the labels of these additional instances.

3.3.2.3.1 Visual Learning

For the operational dataset used in this work, the histogram bars in Figure 3-3(a) highlight the population of flight samples with similar distances from the median flight for each the Pitch Angle variable. An exploration of the histogram bars reveals insights

that help label each group of flights appropriately. The Figure 3-3(b) plot reveals the flights with very different profile from the median flight which is visually represented on the plot in bold red. In most cases, the bar with the highest density or population is often at zero distance from the median flight meaning all flights in that population possess the same profile as the median profile. The range where majority of the flights fall is assumed to represent the normal landing profile for that parameter. The bars closest to the zero-distance bar are often assumed to be a type of normal operation while other bars with the lower density and farther apart from the other bars depict those with abnormal flight profiles. The bars that are farther apart often highlight different flight operational modes from the majority since they have larger distances from the median flight profile.



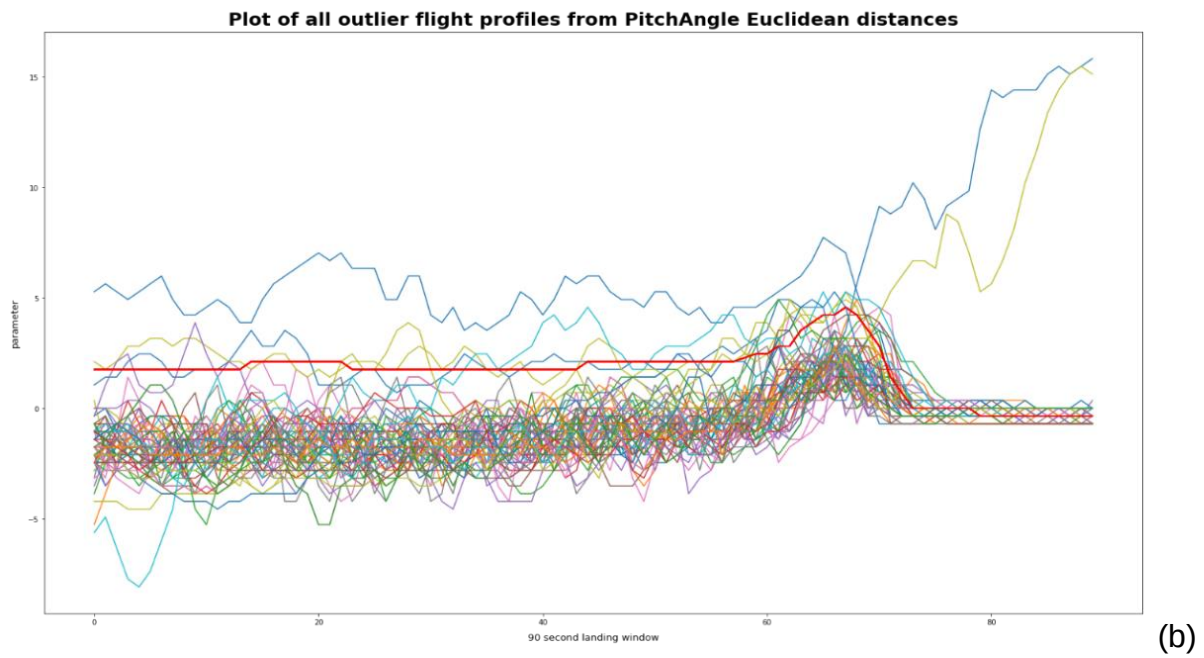


Figure 3-3 Visual representation of flight distribution for Pitch Angle parameter (a) showing density of flights for each profile (b) showing the plot of outlier flight profiles

3.3.2.3.2 Threshold Decision

After gaining appropriate insights from the visualisation, the user is then able to arrive at assumptions that help decide on an optimal threshold for the median-based distance distribution. The threshold point chosen will directly determine what the flight labels would be which in turn affects the accuracy of the labelling approach. Therefore, it is important for the threshold decision to be optimal for the dataset and its domain application.

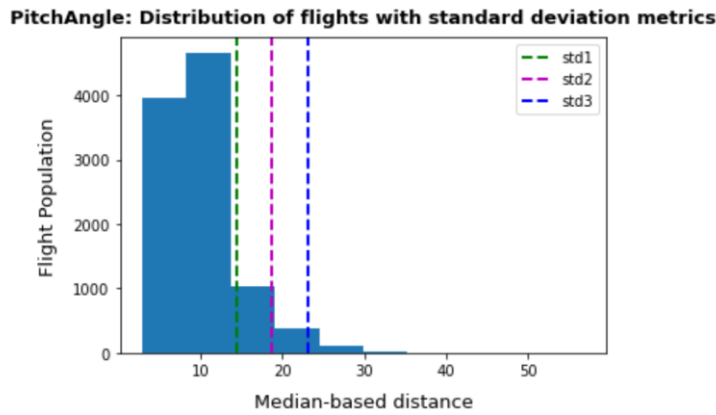


Figure 3-4 Histogram of Pitch Angle flight distribution showing standard deviation metrics

To help determine the threshold, certain statistics of the distribution are extracted. From the histogram in Figure 3-4 above, the green, magenta, and blue dotted lines show the 1-sigma, 2-sigma, and 3-sigma standard deviations respectively. These standard deviation points are iteratively used as the threshold points and coupled with the visualisation; the user can decide on the optimal threshold for that parameter distribution.

Generally, the flights on the left of a threshold would be assigned normal labels while the flights to the right of the threshold line are considered anomalous. For a safety critical domain such as flight operations, it is often more beneficial to set the threshold in a manner that detects as many anomalous events in the flight operations as possible. Misclassifying an anomalous flight as normal could result in being unaware of a hazardous event that could potentially lead to an incident in future. With this consideration, the proposed approach suggests setting the decision threshold as far as possible towards the left while striking a balance between correctly classifying normal flights that are close to the threshold point and capturing low severity or borderline anomalies. The current implementation settled with the 2-sigma deviation point as the threshold for majority of the parameter distributions. Altogether, if a flight

has a distance value above the set threshold in more than one of the parameter distributions, the user labels this flight as a definite anomaly. Where a flight has abnormal profile in only one of the parameter spaces, the insights gained from visualisation helps in the decision making as this example could be an edge of family case that could potentially be classed as either a normal or abnormal flight.

One of the major challenges faced in this work was the labelling of edge cases which are rare occurrences within the dataset. Most machine learning models often ignore these rare profiles especially in highly imbalanced datasets. This was a concern in the current work because of the awareness that these rare cases could signal the presence of precursors to flight incidents which need to be investigated further. Edge cases within the data such as different types of normal or abnormal flight profiles, bring to light the possibility of few flights having multiple labels depending on how the flight profile is understood by the user during visualisation. For example, both Automatic landing (*AutoPilot* engaged) and Manual landing (*AutoPilot* disengaged) are types of normal landing operations that exist in the data and are permitted in the industry. However, a flight where the *AutoPilot* was disengaged well into the 90 second landing window shows a delayed decision was made to manually land the aircraft. This is interesting because it is not normal procedure, and most pilots would have made the decision before the start of this landing window. This could be regarded as an edge or borderline use case which may potentially be assigned either a normal or an abnormal label. While some airlines may treat this as a normal operation since there was no incident, others may like to curb such delayed decisions regarding them as an abnormal operation. Since this work is aimed at revealing every significant abnormal event, the threshold was set to flag as many of these type of cases as possible. This decision was also arrived at after confirmation from experts that it is preferable to misclassify normal flights than to miss a potentially hazardous flight event.

3.3.2.3.3 Validation

Validation of the labelling results is critical as it would be counterproductive to evaluate the methods in the subsequent AD framework against inaccurate target labels from the labelling approach or feed the AD implementation with wrong training data from the labelling workflow. In this study, the NASA data acted as the first approach for validating the proposed median-based method. A domain expert validation was deemed necessary as a second step of validating the results of the proposed labelling approach because it helps increase confidence in the labels from the unsupervised learning and ensuring an accuracy that allows for use in production environments especially safety critical applications. Since it is time consuming for the expert to validate all the flight labels, the flights are grouped according to their flight operating profiles observed during visualisation phase as previously explained. Fifty representative flights from all flight profiles were given to the expert to validate. In this study, it is also assumed that the annotations of the flight instances by the expert are accurate. Similarity labelling was then carried out ensuring that whatever label is assigned to the representative flight, the same label is assigned to all flights in that group assuming that all members within the group will have a high degree of similarity. This final set of labels is then used as target labels for comparing the output from the proposed median-based Euclidean approach. The results are presented and discussed in the next section.

Figure 3-5 below is a workflow that describes how this implementation is used to validate the selected labelling technique. The first validation approach involved carrying out the methods on the NASA dataset. The output labels for the known instances are extracted from all three methods. Domain expert validation was the second way the performance of the proposed method was validated.

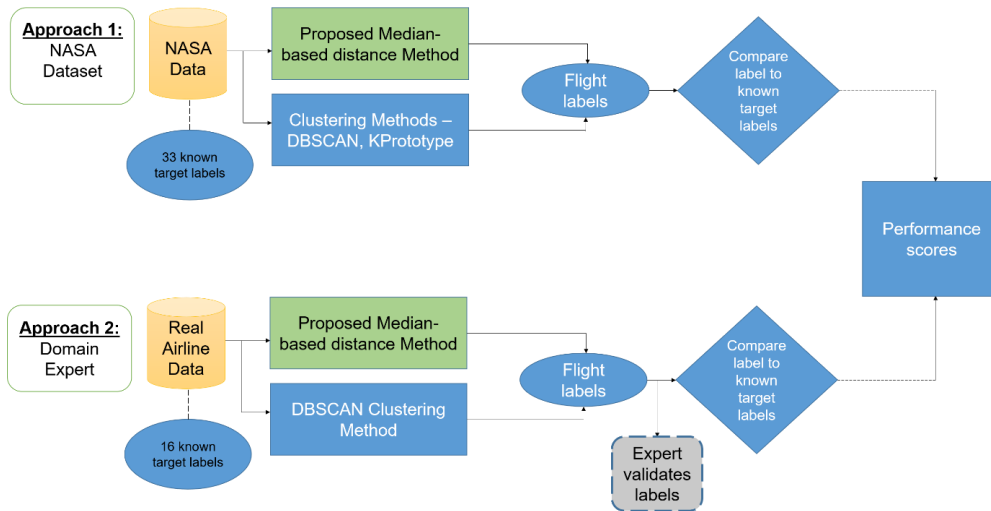


Figure 3-5 Validation of labelling Methods using pre-labelled flight instances

In this case, the better-performing techniques from the NASA implementation are applied to the operational dataset. The performance of these techniques is ascertained by comparing against the target labels from the similarity labelling of the representative instances. Confidently, the label outcomes from the best-performing method becomes the new target labels and ground truth information for the AD implementation.

3.3.3 Labelling Results and Discussions

Of the methods implemented in this paper, the DBSCAN and median-based methods produced the better accuracy in labelling the NASA flights based on the 33 known instances. The summary of the statistical performance of the methods is shown in Table 3-4 below. The result shows that DBSCAN and the proposed method performed equally well on the NASA dataset both labelling 97% of the known instances accurately. In contrast, K-Prototype accurately labelled 69% of the known samples in the NASA data. DBSCAN and the median-based techniques within higher performance are then implemented on the operational airline dataset. The results reveal that the proposed approach outperformed the DBSCAN method by accurately

labelling 94% of the representative instances from the operational dataset while DBSCAN labelled 65% correctly.

Table 3-4 Statistical performance results of DBSCAN, K-Prototype and proposed median-based approach

Method	Labelling Performance on NASA data	Labelling Performance on operational airline data
Proposed Median-based distance method	97%	94%
DBSCAN method	97%	65%
K-Prototype method	69.7%	Not implemented

Contextual Comparisons

In Table 3-5, a comparison is made between DBSCAN and the median-based method showing flight use cases and if both methods correctly or incorrectly labelled them. The use cases are representative examples of the different interesting scenarios observed during analysis. The first use case represents the group of abnormal flights that both methods correctly classified. The second and third flight instances represent abnormal flights which were incorrectly labelled by DBSCAN but correctly annotated by the median-based approach. The group of outlier flights that were incorrectly labelled by the proposed approach but correctly labelled by DBSCAN is highlighted by the fourth flight in the table. The last flight in the table represents the normal flight profiles that DBSCAN incorrectly flagged as abnormal, but the proposed approach correctly labelled as normal.

All abnormal flights DBSCAN labels correctly are those typically with anomalies in their continuous features only. Majority of outlier flights which DBSCAN incorrectly labels

have abnormalities in the categorical variables which it is unable to cope well with. This means that DBSCAN is less effective at correctly labelling flights with anomalies majorly in categorical variables. Since this work is guided by the results of the NASA dataset implementation, DBSCAN appeared to be suitable for application to the operational data. This decision was taken without explicit consideration for the ratio between categorical and continuous variables.

Table 3-5 Representative use cases selected to show interesting profiles and how both methods fared in labelling them

S/N	Flight ID	DBSCAN	Proposed Method	Actual Flight Label	Features with anomalies
1.	106066	✓	✓	Abnormal	PitchAngle, CaptRollPosSSStick
2.	214116	X	✓	Abnormal	Flaps, A/P, AirSpeed, PitchAngle
3.	215518	X	✓	Abnormal	Radio Altitude, RollAngle (deep landing)
4.	104462	✓	X	Abnormal	Radio Altitude only (deep landing)
5.	105815	X	✓	Normal	None

Legend: X means Incorrectly labelled flight; ✓ means correctly labelled flight; CaptRollPosSSStick refers to Roll Angle set from Captain's side stick

While the NASA data comprised of 80% continuous variables, the operational dataset consisted of 67% continuous parameters. As previously discussed, DBSCAN produced more false labels in the labelling of the operational data than the NASA data. This is likely due to the categorical data types being more dominant in the operational data than in the NASA flight data. In other words, anomalies evident in the categorical parameters are ignored thereby misclassifying such flight instances. This confirms that typical distance-based methods may not be appropriate for labelling flight data due to

the existence of mixed data types and the possibility of having a considerably high number of anomalies in categorical variables. It became apparent, during the examination of the incorrectly labelled instances, that the continuous-categorical ratio was a critical factor in the performance of the DBSCAN method. Hence the median-based approach integrated with VIL improves on the labelling results of DBSCAN. Two of the use cases are described below.

Use case 1: Abnormal flight incorrectly labelled by DBSCAN only

This section describes the example of an abnormal flight which DBSCAN incorrectly labelled as normal but which the median-based method labelled correctly. This flight profile reveals a late landing configuration. The graphs in Figure 3-6 and Figure 3-7 comprise four parameter profiles of *AirSpeed*, *AutoPilotEngaged*, *FlapAngle* and *SlatPosition*. Figure 3-6 presents a normal perspective as a basis for comparing the abnormalities in the use case flight displayed in Figure 3-7. Specifically, the *Computed_Airspeed* profile shows the abnormal flight entering the landing window with an airspeed of over 190kt compared to 120kt for the normal flight. The crew obviously struggled to reach the approach speed with the consistent decrease in airspeed until touchdown. The *AutoPilot (AP)* profile shows AP being engaged for an autoland until 50 seconds into the landing window when it is then disengaged for a manual land. An ideal profile should show AP as either engaged or disengaged throughout the 90 second landing window as shown in the AP normal profile in Figure 3-6. Just like the AP, the flaps and slats were also deployed late into the landing window transitioning from lower positions to the expected FULL positions.

Overall, it is expected that the flight crew would have decided on the landing configuration by the start of this window hence indicating an unstable approach and late landing configuration. This use case displays an unsafe sequence of actions, one that would be of interest to flight analysts and should have been labelled as an anomaly by DBSCAN.

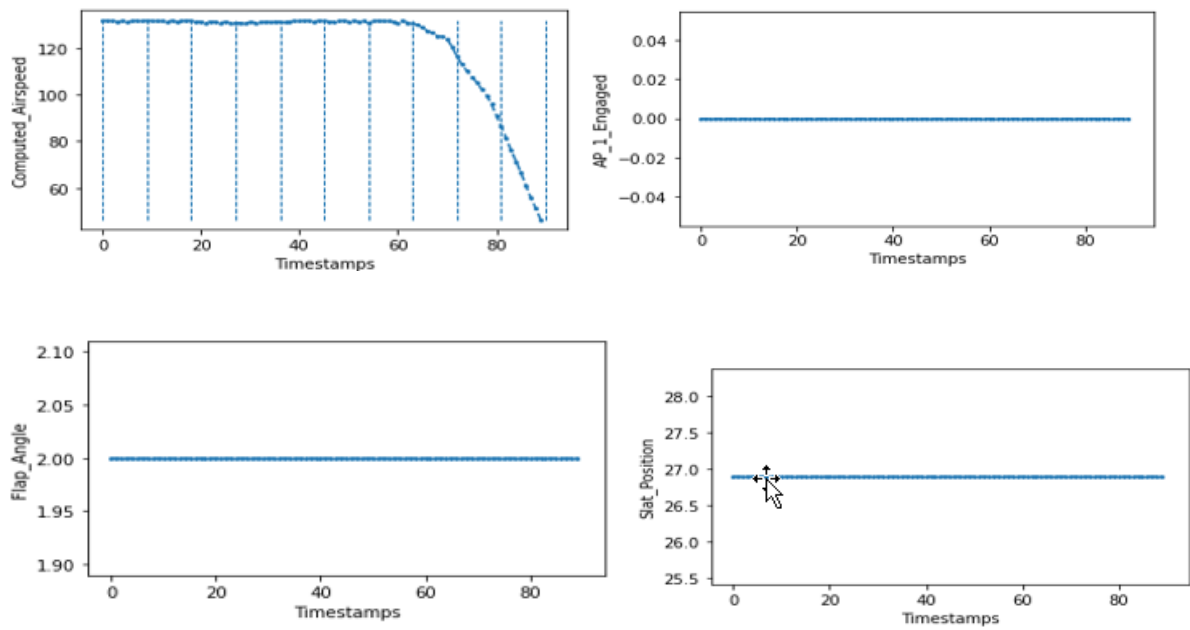


Figure 3-6 Feature characteristics of a Normal flight profile

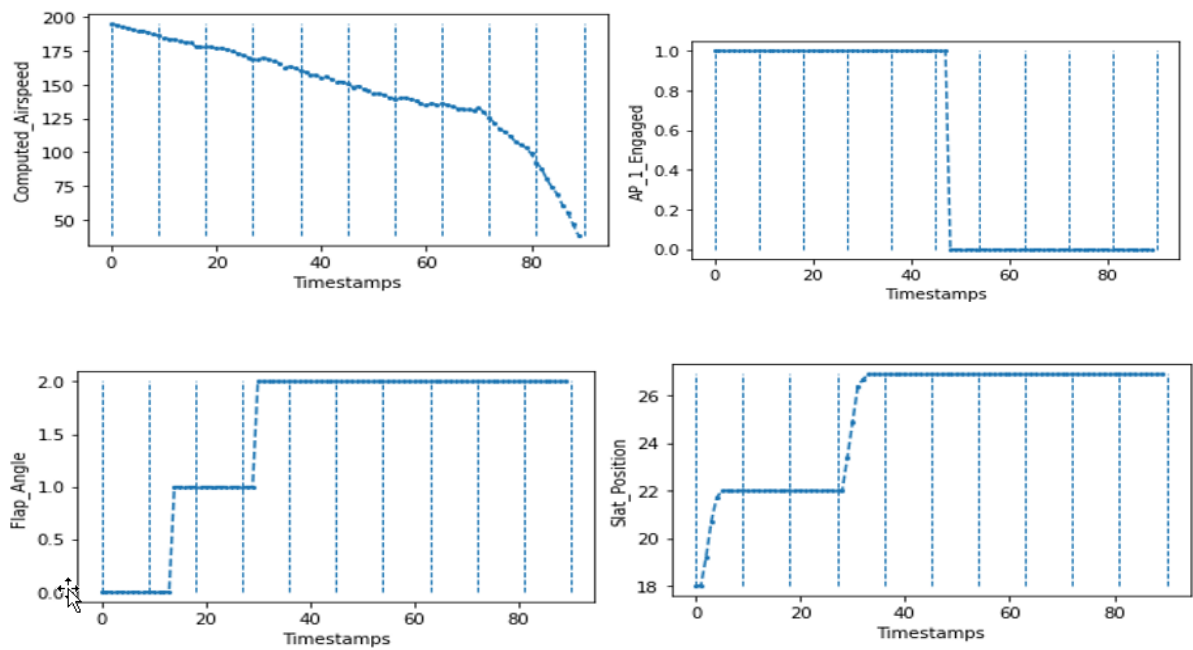


Figure 3-7 Significantly unstable and late landing profile

Use case 2: Abnormal flight incorrectly labelled by Proposed method only

Flight 104462 and 215518 were labelled by the expert as deep landing abnormal flights caused by incorrect actions of the flight crew. Deep landing is a scenario where the aircraft lands at a specific distance behind the glideslope touchdown point. The position of the aircraft is monitored by the glide path as it descends unto the runway. Figure 3-8 presents the Radio Altitude profile of a normal flight showing the aircraft positioned over the threshold point at a height of 50ft above the landing surface. A 3-degree glide angle is ideal in most approaches to landing profiles in this dataset. If the glide angle is smaller, the path is flatter. Deep landing is defined by the slope with reference to the approach. The flatter the approach path becomes, even while maintaining proper height over the end of the runway, the more total landing distance is needed to stop the aircraft which may ultimately lead to runway overrun. Also, the transition phase between final approach and TD referred to as Flare time is assumed at approximately 7 seconds for large aircraft [21].

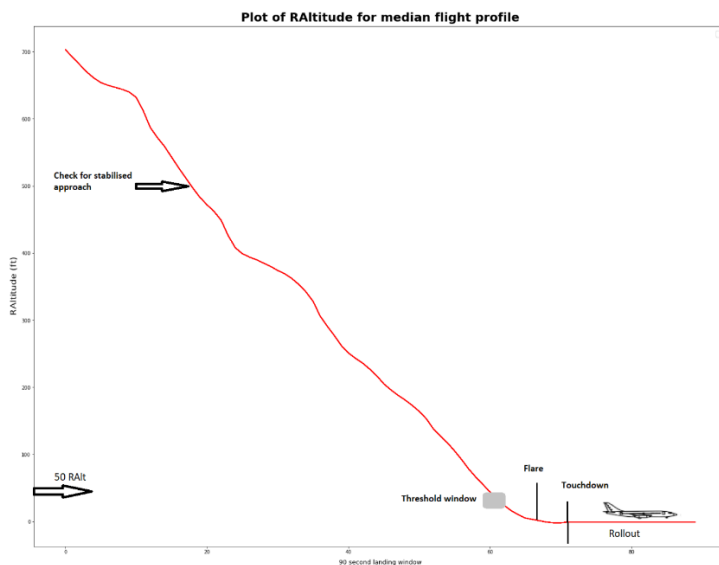
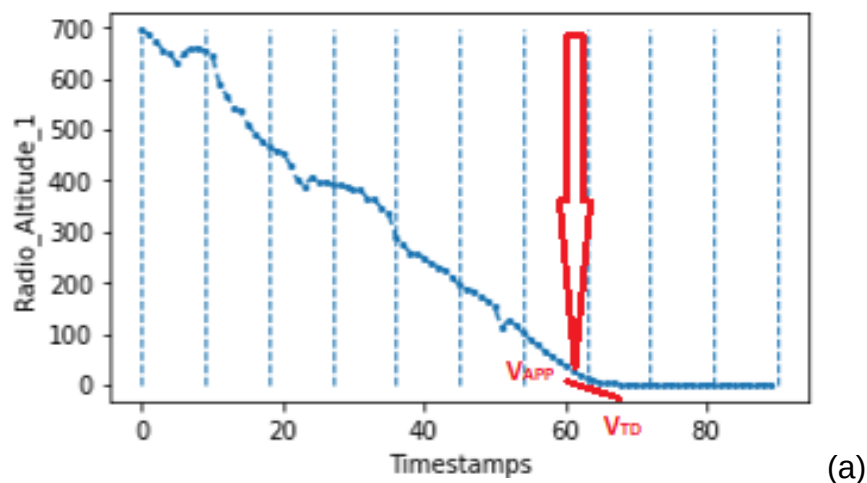


Figure 3-8 Radio Altitude profile for a normal flight showing transition from final approach to touchdown

The deep landing use case in this section is compared to that of a normal profile as shown in the Radio Altitude profiles in Figure 3-8. The V_{APP} symbolises the final approach speed over the landing threshold while the V_{TD} is the Touchdown (TD) speed. 50ft AGL is indicated by the red arrows in the figures. Since deep landing is defined by the glideslope and its glide angle, therefore, the slope is steeper for the normal profile in Figure 3-9(a) than for the slope in Figure 3-9(b) with the abnormal use case. Though it is difficult to ascertain the position of the aircraft exactly from the FDR data, it is possible that the glide angle reduced, and the slope became flatter for the abnormal flight thereby depicting a possible deep landing occurrence which would be of interest to analysts due to the high chance of overrunning the runway.

The proposed approach incorrectly labelled this use case as a normal flight. Lack of evidence in the data is responsible for this as it is only visually identifiable in the Radio Altitude parameter. Experts advise these types of incidents are incredibly difficult to identify due to the resolution of recorded position.



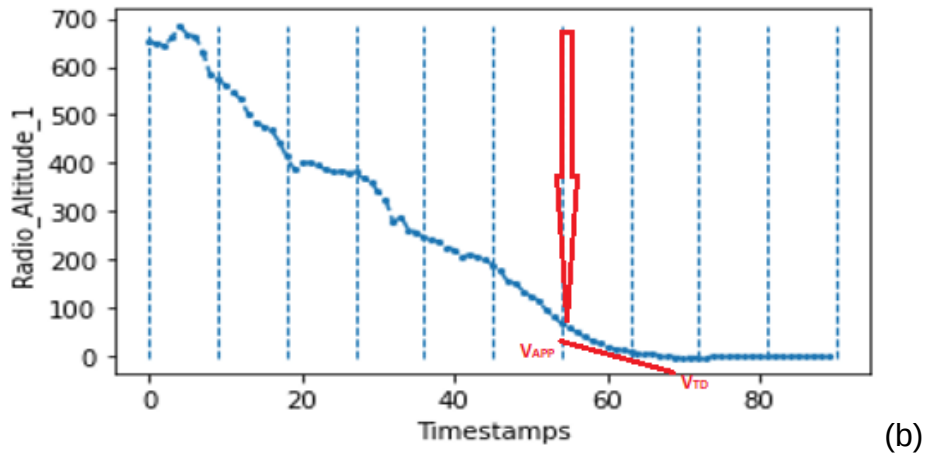


Figure 3-9 Radio Altitude profile of (a) normal profile (b) abnormal use case with deep landing

Some deep landing flights which were correctly labelled by the proposed approach always had abnormalities in more than one feature such as flight 215518 with events in both the *RadioAltitude* and *RollAngle* variables. For the deep landing use case discussed in this section, only the unapparent event in the *RadioAltitude* parameter is used to confirm the flight as abnormal. Figure 3-9(b) also reveals a longer flare time than the normal profile due to the aircraft being behind the threshold point at the 50ft AGL.

The few flights that were mislabelled by the median-based approach have been labelled normal when they should have been labelled anomalies. This outcome first reveals that the approach is more accurate at labelling normal than anomalies. This may be due to the imbalance in the data for the normal class as the model would have many normal training examples from which to learn. Secondly, the flights also show that the threshold set by the user impacts on the accuracy of the labelling process as a threshold too far left would mislabel normal flights and setting the threshold too far right, would mean flights that should be labelled anomalous are mislabelled as normal flights. Nevertheless, the median-based distance technique has an advantage over the DBSCAN method in the labelling outcome when run on both datasets. One of the

reasons for this is because the proposed approach used median as opposed to the popular mean of the distribution for its similarity measurements. Coupled with the univariate strategy imbibed in this method, it is therefore also possible to apply it to the mixed data set while keeping the contextual insights even in the categorical features.

Though clustering techniques like DBSCAN are widely used for classification problems, the labelling requirement is a very important pre-requisite step to flight AD that cannot rely only on clustering for accurate results. Therefore, the proposed approach provides a more robust way of labelling the flight data points with steps like making threshold decisions at parameter level. This approach avoids, to a degree, misrepresentation of data which often stems from projecting high dimensions unto a lower dimensional space [22]. In the proposed approach, the original data, used, eliminates dimension reduction steps thereby reducing the risk of losing useful information[23] and maximum insights are applied for making assumptions and label decisions.

The introduction of visualisation interaction with a user adds an initial validation step to this approach thereby increasing its chances of performing highly in the labelling task. The visualisations generally gave insights on the most representative parameters that help to differentiate normal flights from abnormal ones in addition to highlighting insights about the different operational profiles within the distribution. Although the challenge of dealing with edge cases is a problem faced in both methods, these flights are clearly pronounced in the proposed method than in the DBSCAN method and so, a user can make informed decisions about them that would increase the overall labelling accuracy of the approach. For safety critical applications, it is paramount that the labelling approach is not only accurate, but that assumptions and decisions made are based on clear insight from the data visuals and exploration.

For a larger dataset, there is increased likelihood of more of these edge cases and in turn a possibility of increased inaccurate labels. Therefore, estimating the degree of

uncertainty for the approach is a practical and useful step for stakeholders to assess performance and make final decisions on the labelling outcomes. In section 3.4 of this paper, the estimation of the uncertainty is described.

3.4 Uncertainty Estimation

One of the major challenges of unsupervised machine learning methods in real-world problems is the uncertainty that exists in their outputs due to imbalanced data, the presence of edge cases, feature selection biases and other assumptions made within the data. The presence of these factors in the labelling process reveals a potential level of uncertainty in the approach which needs to be estimated. A domain expert would find it useful for decision making, to know how accurate a model is at labelling and how precise/large the region of uncertainty is. This current work uses parametric uncertainty [24] which measures the uncertainty in each parameter space as described by their probability distributions. The approaches for identifying and evaluating the model uncertainty are discussed in this section but to illustrate the label uncertainty and how it is arrived at, a use case is described in subsequent sections.

3.4.1 Uncertainty Region

Model Uncertainty is inevitable in threshold selection problems, in AD problems and as shown in this study, in the labelling problem. It is an even more challenging task when the data being used is completely or largely unlabelled. In an ideal scenario (without uncertainty), the approach would produce a population of True Negatives (i.e., TN flights correctly labelled as normal) and True Positives (i.e., TP flights labelled correctly as abnormal). In this ideal scenario, there are no False Negatives (i.e., FN flights incorrectly labelled as normal) nor are there False Positives (i.e., FP flights incorrectly labelled abnormal). This current work seeks to estimate the population of false/inaccurate (FN and FP) labels assigned by this method. To achieve this, the thresholds from the labelling phase are used here alongside the distances of the 50 representative flight instances from the domain expert validation. This study introduces

a probabilistic approach. It makes use of the distances of the representative instances to create left and right bounds which form the borders of the uncertainty region and make the uncertainty estimation possible.

To explain how the boundaries are determined, the distribution shown in Figure 3-10 is used. The histogram represents the distance distribution of 10,152 flights for a single parameter as described in the previous section. The bold black line is the optimal threshold point selected in the labelling. To the left of the threshold are flights labelled normal. Out of these normal-labelled flights, it is assumed that those flights closer to the threshold are 'less normal' than those that are farther left of the threshold. The same assumption is made for the flights to the right of the threshold with the 'less abnormal flights being those closer to the threshold point. This indicates that the flights closer to the threshold, on both sides, have a higher probability of being incorrectly labelled. There is therefore a need to ascertain which distance point away from the threshold (on both sides) the incorrectly labelled populations are likely to start and stop within the distribution. The region between these 'start' and 'stop' border points then make up the uncertainty region.

In this study, the boundary points are determined by the median-based distances of the 50 representative flight instances. Their distances are extracted from each parameter distributions and these flight instances subsequently grouped into normal and abnormal flights as labelled by the expert during validation. Among the normal group, the flight with the highest distance or the closest to the threshold is referred to as the 'worst normal'. In the same vein, among the abnormal representative instances, the flight with the closest distance to the threshold or smaller distance measure than the other flights in the abnormal group, is referred to as the 'best abnormal' flight.

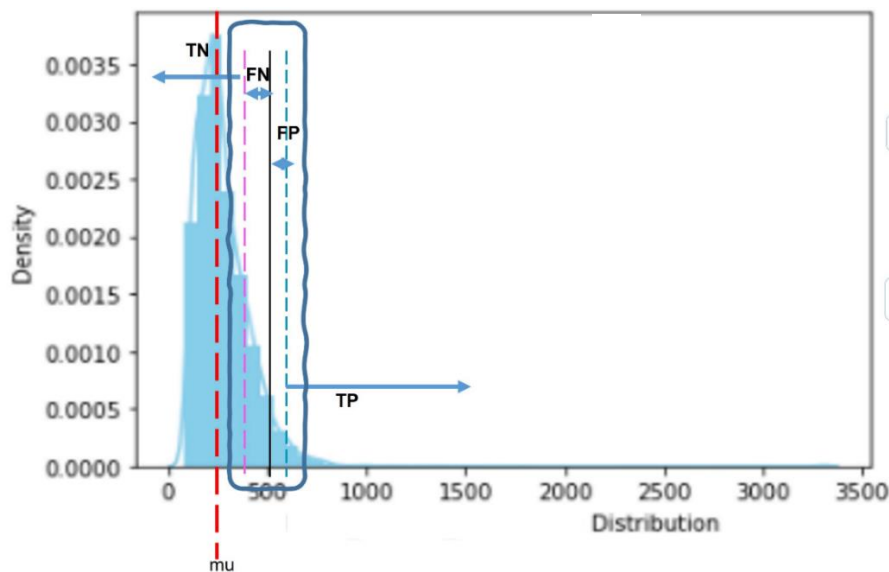


Figure 3-10 The univariate distribution describing the uncertainty region of the labelled data

The ‘worst normal’ or left-bound point (magenta dotted line) sits between the mean of the distance distribution (red-dotted line) and the threshold point. All flights to the left of the left bound point are grouped as flights with TN labels meaning that they are obvious normals while the flights to the right of the left-bound point are depicted as having a higher possibility of being FNs or inaccurately labelled as normal. The right-bound point is determined in a similar fashion but with an additional condition that the right-bound must also be greater than the threshold point. This is necessary as there are some of the representative instances that may fall within the normal region too. In other words, these flight profiles cannot be regarded as abnormal for that parameter hence are not considered during selection of the right-bound point. Based on the premise that the closer an anomalous point is to the threshold line, the closer it is to being normal. The ‘best abnormal’ point on the right of the threshold point is then chosen as the right-bound thereby separating the population of flights with a higher probability of being abnormal (True Positives) from the population with a high likelihood of being inaccurately labelled as abnormal (False Positives).

Therefore, the entire uncertainty region falls between the right and left bound points and the probability of flights falling into this region provides a measure of uncertainty for the labelling decisions made for that parameter space. The probability of inaccuracies is determined by the ratio of the sum of the false label population to the total population. The same process is carried out for all the parameters and the worst, that is, the parameter with the highest uncertainty is selected as the level of uncertainty for the proposed labelling approach. This produced a 12% uncertainty for the labelling approach given the dataset and the representative instances used.

3.4.2 Uncertainty Estimation Results and Discussion

With the uncertainty calculation, the probabilities of uncertainty are recorded when iterated over each parameter. The features with the highest probability of inaccurate labels is the Airspeed and PitchAngle both with a 12% uncertainty directly higher than the RAltitude with 8.8% uncertainty. Figure 3-11 and Figure 3-12 reveal the uncertainty region shown in amber colour for the parameters with the highest labelling uncertainty.

Airspeed profile has a wider range of values just like PitchAngle. That shows that the airspeed used for landing at this airport varied considerably across all the flights. This contrasts with categorical features which produced very small uncertainties such as FlapAngle with an uncertainty of 0.3%. Being a safety critical application, the worst uncertainty from the Airspeed distribution is chosen as the overall estimated uncertainty possible when using this labelling method for the flight dataset. In other words, it can be concluded that the proposed median-based labelling method is unsure about the annotations given to 12% of the population of the data.

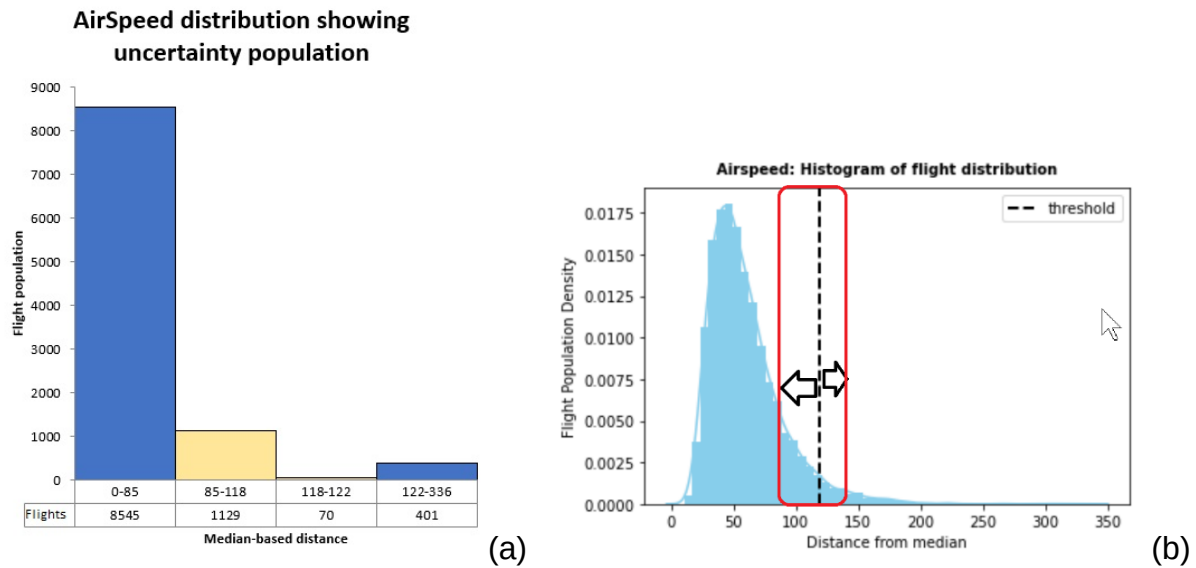


Figure 3-11 Airspeed distribution showing uncertainty (a) Uncertainty population in yellow (b) Uncertainty region in red

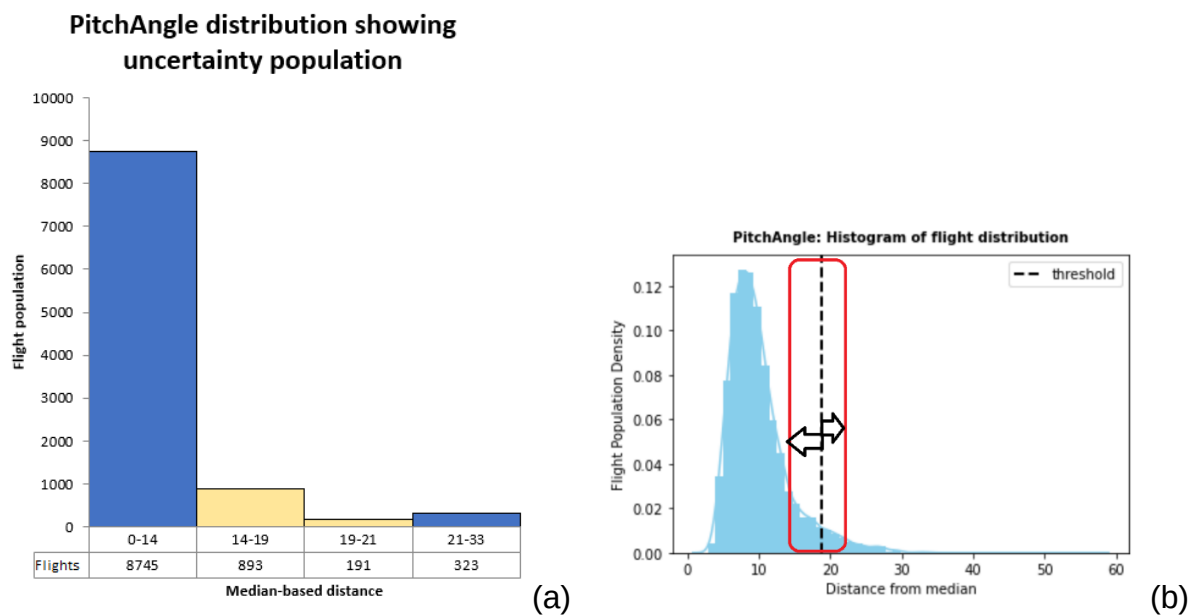
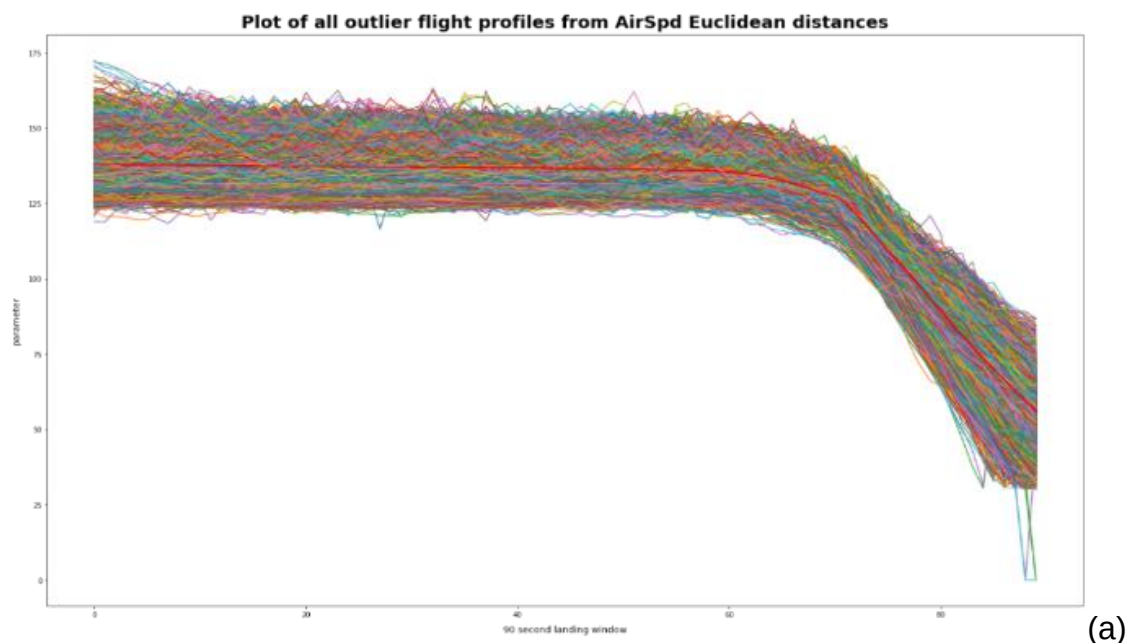


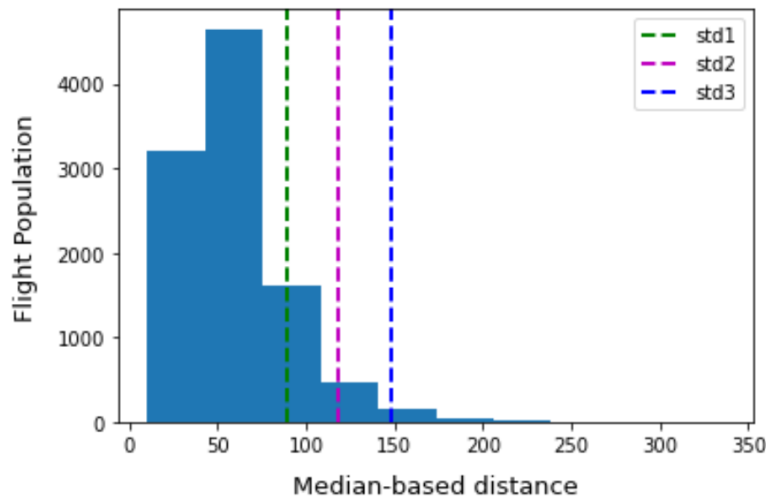
Figure 3-12 PitchAngle distribution showing uncertainty (a) Uncertainty population in yellow (b) Uncertainty region in red

Visualising features like the Airspeed showed that there were differing operational choices for airspeed adopted by pilots for landing the aircraft. Also, the Airspeed

histogram revealed relatively high population of flights than some other features. For example, Figure 3-13(a) below shows majority of the flights entered the landing window with the median airspeed of about 137kt from the final approach which is equivalent to the normal profile represented by the first two bars in Figure 3-13(b). Then, over 1500 flights comprising the third histogram bar make up the population of flights with Airspeeds about 1-sigma (represented by the green dotted lines) from the median speed and though slightly dissimilar to the median speed profile, are still depicted as having a normal speed profile. However, over 500 flights represented by the fourth distance bar still had airspeed values that were less than 3-standard deviations (3-sigma) from the median profile and it is at this point that labelling uncertainty begins to set in and the user asks questions like: *are these flights normal, edge of family or abnormal?* With this parameter distribution, the region of uncertainty between the left and right bounds of the uncertainty calculation will always have a relatively higher population than in some other feature space due to the large number of flights that fall into the different bars or use the different landing airspeed configurations. The higher the population in the uncertainty region, the higher the probability of uncertainty in that feature space.



Computed_Airspeed: Distribution of flights with standard deviation metrics



(b)

Figure 3-13 Plot of landing flight profiles with the median profile in bold red. 13(b): Histogram visualisation of flight distribution for AirSpeed feature

3.5 Conclusion

This paper discussed the importance and performance of unsupervised labelling of flight data to distinguish normal from abnormal flights as a critical step for flight anomaly detection. The proposed approach uses a median-based distance/similarity measure and integrates this with visual learning by a user who makes a judgement call on the decision boundary based on the visual information presented. In this approach, it was shown that the framework can produce results that are more intuitive than many other techniques. The median-based approach with user visualisation performs better than normal clustering techniques on operational flight datasets. While the latter provided an 65% accuracy, the proposed approach was able to label the flights with a 94% accuracy. Also carried out, was an estimation of uncertainty in the outcome of the proposed labelling approach. By creating upper and lower boundaries for the distribution, the estimated uncertainty of 12% was arrived at for the flight dataset used. The probabilistic uncertainty estimation method is used because it is a

well-established concept and the median-based distance approach, because classification applications like labelling are usually addressed with such statistical distance measures. Therefore, it is expected that this labelling framework can also produce good results on other flight phases or on a more generic dataset that is not tailored to a particular flight route as evidenced with the implementation on the NASA dataset.

References

1. Chandola, V., A. Banerjee, and V. Kumar, *Anomaly detection: A survey*. ACM computing surveys (CSUR), 2009. **41**(3): p. 15.
2. Igenewari, V.R., Z. Skaf, and I.K. Jennions. *A Survey of Flight Anomaly Detection Methods: Challenges and Opportunities*. in *Proceedings of the Annual Conference of the PHM Society*. 2019.
3. BEA, *On the accident on 1st June 2009 to the Airbus A330-203 registered F-GZCP operated by Air France flight AF 447 Rio de Janeiro - Paris*, in *Safety Investigations*. 2012, Bureau d'Enquêtes et d'Analyses: France.
4. Das, S., et al. *Comparison of algorithms for anomaly detection in flight recorder data of airline operations*. in *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. 2012.
5. Theissler, A., et al. *VIAL-AD: Visual Interactive Labelling for Anomaly Detection-An approach and open research questions*. in *4th International Workshop on Interactive Adaptive Learning (IAL2020)*. 2020.

6. Felder, R.M. and R. Brent, *Active learning: An introduction*. ASQ higher education brief, 2009. **2**(4): p. 1-5.
7. Bernard, J., et al., *Comparing visual-interactive labeling with active learning: An experimental study*. IEEE transactions on visualization and computer graphics, 2017. **24**(1): p. 298-308.
8. Settles, B., *Active learning literature survey*. 2009.
9. Holzinger, A., et al., *Interactive machine learning: experimental evidence for the human in the algorithmic loop*. Applied Intelligence, 2019. **49**(7): p. 2401-2414.
10. Beil, D. and A. Theissler. *Cluster-Clean-Label: An interactive Machine Learning approach for labeling high-dimensional data*. in *Proceedings of the 13th International Symposium on Visual Information Communication and Interaction*. 2020.
11. Hall, P., D. Titterington, and J.-H. Xue, *Median-based classifiers for high-dimensional data*. Journal of the American Statistical Association, 2009. **104**(488): p. 1597-1608.
12. Wordpress. *Medians in high dimensions*. Statistical Odds a& Ends 2018 December 21, 2018 [cited 2021 10.10.2021]; Available from: <https://statisticaloddsandends.wordpress.com/2018/12/21/medians-in-high-dimensions/>.
13. Solomatine, D.P. and D.L. Shrestha, *A novel method to estimate model uncertainty using machine learning techniques*. Water Resources Research, 2009. **45**(12).
14. Grimmeisen, B. and A. Theissler. *The machine learning model as a guide: pointing users to interesting instances for labeling through visual cues*. in

Proceedings of the 13th International Symposium on Visual Information Communication and Interaction. 2020.

15. (NASA), N.A.a.S.A., *Multiple Kernel Anomaly Detection (MKAD) sample dataset*. 2013.
16. Ester, M., et al. *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*. 1996. Portland, Oregon.
17. Schubert, E., et al., *DBSCAN revisited, revisited: why and how you should (still) use DBSCAN*. *ACM Transactions on Database Systems (TODS)*, 2017. **42**(3): p. 1-21.
18. Zhao, W., et al., *An Incremental Clustering Method for Anomaly Detection in Flight Data*. arXiv preprint arXiv:2005.09874, 2020.
19. Huang, Z. *Clustering large data sets with mixed numeric and categorical values*. in *Proceedings of the 1st pacific-asia conference on knowledge discovery and data mining,(PAKDD)*. 1997. Citeseer.
20. Huang, Z., *Extensions to the k-means algorithm for clustering large data sets with categorical values*. *Data mining and knowledge discovery*, 1998. **2**(3): p. 283-304.
21. Duven, J.E., *Landing Performance Data for Time-of-Arrival Landing Performance Assessments*, U.S.D.o.T.F.A. Administration, Editor. 2015, U.S. Department of Transportation Federal Aviation Administration: US. p. 10.

22. Wójcik, P.I. and M. Kurdziel, *Training neural networks on high-dimensional data using random projection*. Pattern Analysis and Applications, 2019. **22**(3): p. 1221-1231.
23. Nanduri, A. and L. Sherry. *Anomaly detection in aircraft data using Recurrent Neural Networks (RNN)*. in *2016 Integrated Communications Navigation and Surveillance (ICNS)*. 2016. Ieee.
24. Pulkkinen, U. and T. Huovinen, *Model uncertainty in safety assessment*. 1996.

4 Ensemble Anomaly Detection Framework for Aircraft Operations

The standard technique used by airlines for highlighting flight of potential interest employs a set of rules applied to the onboard data. Its performance is dependent on how well the rules are designed but generally fail to find unknown anomalies. Therefore, several Anomaly Detection (AD) algorithms like Multiple Kernel Anomaly Detection (MKAD) have been developed to improve on the performance of conventional methods by finding usually unknown anomalies in FDR data. Limitations of these individual methods in terms of their inability to detect anomalies in specific data types, has given rise to the need for more robust approaches for AD. For flight AD, the development of a robust automated anomaly detection framework is critical for aviation safety especially considering the expected increase in air traffic in coming years. To meet this industry need, an Ensemble AD is proposed in this work to produce a wider array of anomalies than the individual AD methods that constitute the ensemble. Although there are several ensemble techniques proffered in literature for classification and clustering problems, there are however much less typical ensemble-based methods proposed for anomaly detection especially in the aerospace domain. In this study, MKAD has been chosen as the baseline algorithm since it has become the state-of-the-art technique for unsupervised flight AD and was developed by NASA. This paper implements some flight AD algorithms and proposes an ensemble framework for combining the output of the individual AD algorithms. The proposed method shows some improvement over and above in finding abnormal flight events that were missed by the baseline method.

4.1 Introduction

With air travel expected to double by 2035 [1], the likelihood of safety-related risks also increases for the industry. There is therefore a need to be innovative about proactively identifying operationally significant safety risks that are currently not being monitored. Flight risks can be classed (based on severity levels) as accidents, incidents, events and precursors [2]. An **accident** refers to an occurrence on an aircraft where an individual suffers severe injury, death or where substantial damage is inflicted on the aircraft. An **incident** is an occurrence that is not an accident, but which affects or could affect the safe operation of the aircraft [3]. On the other hand, an **event** occurs when a flight parameter exceeds a previously set safety threshold but does not is not considered unsafe operation until validated by an expert. Factors such as whether that, either alone or in combination with other factors, contribute to the increased likelihood of risk occurring are referred to as **Precursors**. Accurate diagnostic capabilities of detection systems can help in the core step of identifying these vulnerabilities for eventually mitigating risks. The mitigation tasks could be in the form of reviewing safety procedures in view of the identified anomalous events, improving training, and enhancing system designs. In addition to detection systems taking false alarms into consideration in the requirements phase of development, they must also be careful not to miss real abnormalities which particularly affect flight safety.

The aviation industry can identify these risks from sensor data within varied systems on an aircraft. The data takes the form of measurement recordings of systems on each flight. These recordings from the variety of systems cover quantities such as the positioning of the aircraft, engine parameters, pilot system inputs and autopilot modes. The data in its time series form is analysed by flight safety experts who make use of domain knowledge and airline policies to identify known events that are operationally significant.

The state-of-the-art method currently used by airlines in the industry is referred to as the exceedance approach. It highlights pre-determined events within a fixed list of parameters and flags these events when they exceed set thresholds. The problem with this approach is that interesting events that have not been specified apriori will likely be missed by this rule-based approach. In a safety-critical environment such as airline flight operations, missed anomalies could be catastrophic. A report by the Federal Aviation Administration (FAA) [4] highlights that the current exceedance-based detection system lacks advanced analytical abilities thus requiring improvement. Improvement efforts would increase the possibility of finding new vulnerabilities or threats. Enhancing flight safety will entail developing, testing, and validating Machine Learning (ML) approaches that have the potential of automatically identifying anomalous events within the data and highlighting the contributing factors that can be subsequently addressed. To be trusted and their outcomes acted upon by the stakeholders, such methods are expected to produce lower false alarms as well as lower number of missed events.

Apart from the current exceedance detection method, other approaches have been developed for automatically detecting anomalies within observed data to ensure flight safety. While some of these techniques have been model-based, recent methods are focusing on data-driven approaches in line with the goal of the industry to continue to use data to inform such decision-making. Data-driven approaches, unlike model-based approaches, do not need prior knowledge of the system behaviour but make use of historic data to model the system behaviour and provide detection abilities. **Anomaly Detection** focuses on finding instances of the input data that differ from the general behaviour that characterises most of the data instances. Flight AD provides a powerful means of identifying the anomalous events as well as trends that reduce safety margins in the aviation domain.

Flight AD techniques are data-driven approaches that are aimed at finding previously unknown operational abnormalities within flight data; a capability that is a limitation of

the conventional exceedance-based approach. MKAD [5] and ClusterAD[6, 7] are examples of such AD methods. A comparison [8] carried out on these two techniques revealed limitations of the individual algorithms. The work highlighted that MKAD was better at finding anomalies in categorical features within the FDR while ClusterAD performed better at finding abnormalities of flights present in continuous features. In aircraft operations, this means that implementing each of these individual AD methods alone could potentially miss critical safety events that the other is likely capable of detecting. The drawbacks of these individual methods demonstrate that there is a need to develop a robust AD framework with a higher accuracy in detecting anomalies in heterogeneous data types. Motivated by these challenges, this paper proposes a data-driven ensemble AD that combines the outcomes of individual data-driven choice detectors to form a robust approach to finding a wider array of anomalies than the baseline or individual techniques.

4.2 Related Work

Various efforts have been put into enhancing aviation safety over the years. This section reviews some of the flight anomaly detection methods described in literature. In the aerospace domain, several algorithms have been developed to address the problem of AD and recent reviews [9, 10] summarise how the identification of anomalies is carried out for each. Some detectors rely on clustering methods, such as the Sequence Miner algorithm[11] for discrete flight features identifying flight level anomalies and the Inductive Monitoring System (IMS) [12] for continuous parameters only. These studies are based on determining “normal” regions in the feature space, quantifying the distance between each flight instance from the normal regions and then calculating an anomaly score. Orca [13] uses Euclidean distance of a point to its k-nearest neighbours to ascertain the degree of novelty of each flight instance. It is

however limited in that it loses the temporal aspect of abnormal flights because it tackles each flight independently.

Another study for flight anomaly detection developed the MKAD algorithm [5]. MKAD combines the strength of both vector space and sequential AD techniques to allow a wider range of anomaly detection. MKAD is useful for discrete data due to its use of the normalized Longest Common Sub-sequence (nLCS) kernel. In this case, the nLCS refers to the length of the longest sub-sequence that is common to the new flight and the normal flight sequences. The ability to use information from both data types makes it possible for MKAD to identify more complex behaviours in the AD process than previous methods mentioned. It is, however, able to find anomalies in categorical parameters more than it can detect those present in continuous features. This is likely due to loss of some finer continuous features when continuous data is discretised. For example, while it was able to detect anomalies in an auto-landing state due to an autopilot mode anomaly, it was unable to detect a high-speed approach evident in continuous parameters [8].

Two forms of ClusterAD are found in literature developed to find different types of anomalies. The ClusterAD-Flight method [14] is designed to find pattern based anomalies. It transforms time series flight data into high-dimensional vectors, ensuring different flights are comparable using distance-based measures from an anchoring event and then introducing the density-based spatial clustering of Applications with Noise algorithm (DBSCAN). A potential limitation of this method can be that it isolates each time sample in each signal as a unique feature, ignoring the change over that time which may be important. ClusterAD, generally, is tuned to find anomalies in FOQA type data but does not clearly define the similarity measure of discrete data thereby producing a low performance in finding anomalies in discrete data. In other words, it is not sensitive to the sequences of categorical features since these parameters are processed in the same way as continuous variables hence, only state differences are observable by the algorithm.

ClusterAD-Sample [7] is the other form of ClusterAD built to find instantaneous anomalies in flight data. It undergoes a transformation step as well converting raw multi-variate time-series flight data into vectors such that each flight instance at a specific time is represented by a vector. Then normalising each flight feature is carried out. In this reduced form, a clustering method that is based on Gaussian Mixture Model (GMM) is applied on the data. GMM is a probability density function (pdf) computed from the weighted sum of gaussian component densities where each component represents a frequently observed flight operation such as touchdown. The normal flight operation is thus determined with GMM, and detection of anomalies is determined based on the calculated pdf scores for each instance. The comparison review [8] carried out on these techniques shows that ClusterAD-Sample found more severe anomalies than MKAD and ClusterAD-Flight. This is possibly because its cluster analysis is based on data samples of the flights and due to its ability to discover instantaneous anomalies, can find more anomalies.

Despite these potential limitations, MKAD and ClusterAD represent the recent state-of-the-art for automated flight anomaly detection today as they have proved to be very effective at uncovering several anomalies that are missed by conventional exceedance or rule-based approaches. From the comparison made in a study [8] about both methods, the authors proffered that combining both techniques would mean combining their strengths at finding anomalies in different data types thereby complementing their weaknesses. This work is motivated by the concept of combining these methods and hence, proposes an ensemble anomaly detection framework. The ensemble AD brings together the strengths of few choice detectors to produce a more robust algorithm with an improved performance than the baseline detector thereby increasing the possibility of finding more and a varied range of true anomalous situations.

4.3 Ensemble Anomaly Detection Framework

An Ensemble AD is a combination of two or more AD methods to form one method that is expected to outperform even the sophisticated individual approaches used for AD. There have been different applications of ensembles for AD [15]. Ensemble-based algorithms for anomaly detection combine results of multiple detection models using different combination techniques. This section of the work presents the ensemble framework overview, the data used and the initial data preparation steps.

4.3.1 Overview of Framework

Figure 4-1 describes the proposed Ensemble framework for anomaly detection in the aviation domain. The figure reveals the data pre-processing and labelling of the raw onboard flight recorder data as pre-requisite steps for the proposed Ensemble AD phase. The pre-processing step is essential to remove noise, select relevant features and transform the data while preparing it to be used by the ML models. This ensures that the models can learn appropriately, the characteristics of normal behaviours, with a view to accurately determine anomalous instances. Directly preceding the anomaly detection phase is the Labelling phase. In the previous chapter, a labelling approach was implemented for labelling the partially labelled flight data available for this work. The input into the labelling phase is the prepared data and the output is a set of labels for every flight instance in the dataset.

The Ensemble AD phase then trains its individual detectors on a set of training data from the labelling phase, merges their outcomes using a combination technique and produces a final set of predicted anomalies. A domain expert is then employed to validate these anomalies to ascertain their operational significance. The output of the ensemble is also compared with the target labels from the labelling phase generating a confusion matrix that summarises the statistical performance of the ensemble framework compared to the baseline method.

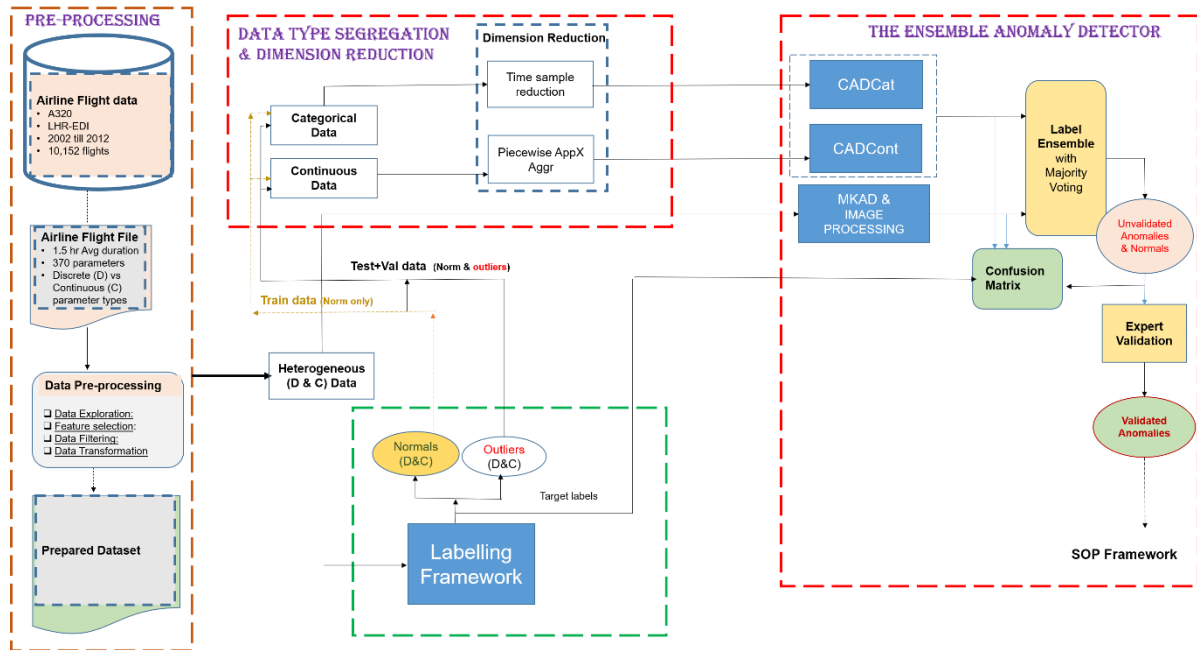


Figure 4-1 Proposed Ensemble Anomaly Detection Framework for Flight Operations

4.3.2 Dataset and Data Preparation

As described in chapter 3, the FDR dataset used in this study was collected from flights flown between 2002 and 2012. It contained 10152 A320 flights on one flight route but various runways at the departure and arrival airports. Each raw flight data included 370 parameters covering engine parameters, speeds, radio altitudes, switch modes and winds. Flights that share a common route were used to ensure the flight profiles can be compared. For the transformation step, flight instances were extracted at 1-second intervals from 70 seconds prior to touchdown to 20 seconds into roll covering the final approach to landing phase. This means that the same number of samples i.e., 90 was obtained for all the flights. After performing the correlation and feature selection with the help of a domain expert, the number of features was reduced from

370 to 48. Therefore, the dimension of the data was reduced to 48 features * 90 samples.

In the initial stages, the 48 features were explored for all 10152 flights thereby providing useful insights into the functionality of each feature. The exploration step also revealed that the categorical features could be reduced in a further dimension reduction step by having a 3-sample rather than the initial 90-sample feature space. This further dimension reduction step is explained in section 3.4.4. The 48 features comprised of 32 continuous parameters and 16 categorical parameters and are listed in the Table 4-1 and Table 4-2 below. Worthy of note is the fact that the flap positions were transformed into categorical variables by binding the angles into three values: 0 (0-15), 1(15-34) and 2 (>=35 degrees).

Table 4-1 Table of 32 continuous features in flight data

32 Continuous features			
LongitudinalAcceleration	SlatPosition	SideStickCaptainPitch	NormalAcceleration
CaptainRollSS	WindSpeed	SidestickFORoll	LateralAcceleration
SpoilerLH4	LH5Spoiler	SpoilerRH2	RH5Spoiler
RH3Spoiler	2LH5Spoiler	VMAXOperatingAirspeed	RudderPosition
TrackSel	SelectedVS	AllSpeedAileronPosLeft	RgtAileron
HeadingSel	FltPathSel	DecnHeightSel	AltiSel
ApproachSpeedTarget	ThrottleAngleEng_1	ComputedAirspeed	RollAngle
PitchAngle	RadioAltitude1	GroundSpeed	FOPitchSS

Table 4-2 Table of 16 categorical features in flight data

16 Categorical features			
AutoP2Eng	AP1Engaged	ATSActive	GndSpoilerArmed
EngagedATS	BrakeOnOffAutomatic	InLandingConfigConf3	x1OutSpoilrL
GNose	RevDepldEng2	ReverserDeployedEng1	GLeft
RSpoilOut	SpeedBrakeCommand	GRight	FlapAngle

With exception to 16 flights, the 10152 flights were originally unlabelled but a median-based approach with Visual Interactive Learning capability was applied to assign labels to them in the labelling phase as described in chapter 3. This preceding phase labelled 702 out of the 10152 flights as anomalous. The previous work carried out on this study also proposed a method for computing the uncertainty in the labelling approach resulting in a 12% uncertainty. For effective training and testing of the detection models, a subset comprising of 1856 flights was used for the remainder of the study. This ensured that only those flights in the core of the distribution with a high label confidence are used subsequently for effectively training the models in this proposed approach. The constituents of the 1856 core subset of flights were selected to resemble real-life bias within the original data i.e., biased with high proportion of normal to anomalous flights. The core subset data becomes the input into the choice detectors of the ensemble.

4.4 Individual detectors

In this work, a few AD techniques were considered to make up the constituent detectors for the ensemble AD framework. This section of the study describes the setup process for each detector, the input as well as the expected output. There are

post-implementation steps that are common to the three choice detectors selected for this study. After the three models are implemented, they produce a probability of normalcy score for each flight instance. The probability scores tell how normal the model ranks each flight instance. For example, in the Cumulative Distribution Function (CDF) plot in Figure 4-2, the scores range from 0.5 to 0.85 meaning that flights with scores around 0.85 are regarded by MKAD as being normal while those closer to 0.5 have severe anomalies in them that MKAD has been able to detect. Therefore, the lower the score, the more anomalous the flight instance.

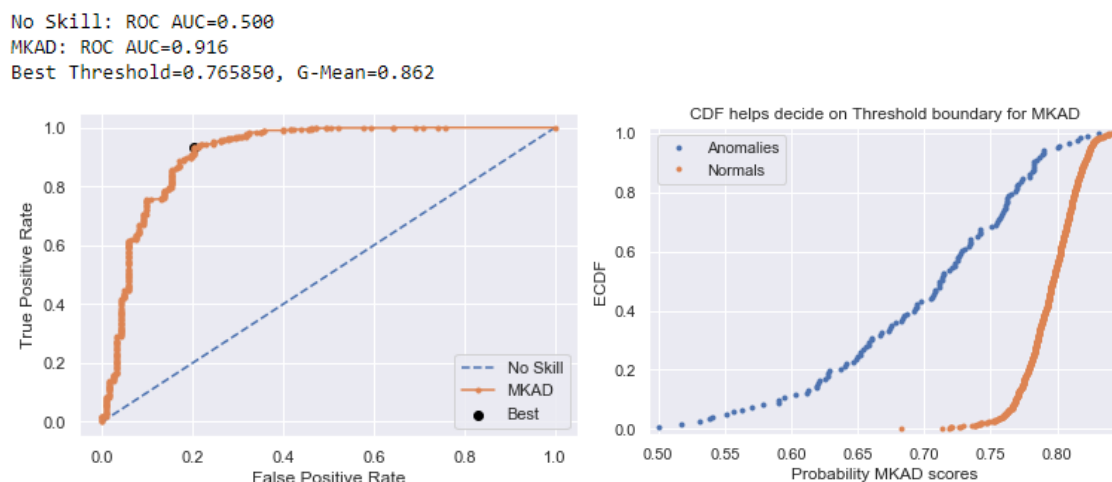


Figure 4-2 Example of (left) ROC and (right) CDF plots for determining optimal threshold point for detector model

It is then critical to know at which point, in the distribution, the decision boundary should be drawn to determine which of the two classes a flight would belong. A Receiver Operating Curve (ROC) is a metric constructed with a false positive rate (FPR) and true positive rate (TPR) on the x and y axis respectively. This means that the top left corner of the ROC plot (0,1) is the “ideal” point i.e., zero false alarms and all anomalies detected. Although unrealistic, it does mean that a larger Area Under the Curve (AUC) is usually better. ROC is typically used in binary classification to explore

the output of a classifier. The true labels and the probability scores form the input variables into the python 'roc_auc_score' function used here. The optimal threshold of the scores is also estimated within this function. The plot ascertains the classification of the flights by varying the threshold settings until it reaches an optimal that strikes the best balance between the probability of detecting a true anomaly and the probability of a false alarm. The ROC curve shown in Figure 4-2 reveals the best/optimal threshold for the model as well as the AUC for the model predictions compared to the AUC for predictions of a basic model with no learning skill at all. The CDF plot in Figure 4-2 provides us a visual way of validating the threshold point computed by the ROC. The threshold is then applied to the test/validation datasets (with new instances unseen by the model) for each model to ensure it produces sensible results on new data instances.

4.4.1 Individual detector: MKAD as baseline

MKAD is developed to work on heterogeneous data. The discrete binary switch sequences are used in building a discrete kernel and discretized continuous parameter features are used to form the continuous kernel for MKAD. MKAD combines these multiple kernels into a single optimization function using the One Class-Support Vector Machine (OC-SVM) framework. It also needs nominal or near nominal data to learn proper hyperplanes. In this algorithm [16], the OC-SVM constructs an optimal hyperplane that distinguishes normal samples in the kernel by maximizing the margin between the origin and the hyperplane. The weighting ratio used in this work for discrete and continuous data types is 0.6 to 0.4 respectively since there are more continuous features in the data. Settings such as this for the MKAD engine influence the types of anomalies detected therefore, this work varied them to get an appropriate outcome. The pdf of all flight instances is extracted as the outcome scores from MKAD. With an optimal threshold of 0.77, the ROC in Figure 4-2 shows that MKAD correctly classifies 92% of flights when implemented on the training dataset. The CDF gives a

clear view of MKAD performance at a 0.77 decision boundary; more false alarms than missed anomalies.

Apart from probability scores, MKAD also provides additional insight into the anomalous events that contributed to a flight instance being classed as anomalous. A simple postprocessing method is used to rank the anomalous flights and decompose the anomaly score of individual flights in terms of discrete and continuous parameter contributions. Table 4-3 below is an extract from the MKAD output that lists the anomalous flights detected categorised according to the data type in which anomalies were discovered as well as how they rank in terms of severity. Each anomalous flight details information about features within it that contain anomalies. This will be described in more detail in section 4.7.2 when the results and use cases are explained.

Table 4-3 MKAD list of anomalous flights categorised by the data type with the anomaly

Continuous	Discrete	Continuous + Discrete
Rank: 35 Flight ID: fltID_255417_3986	Rank: 21 Flight ID: fltID_257553_4190	Rank: 1 Flight ID: fltID_246040_3134
Rank: 36 Flight ID: fltID_366273_6050	Rank: 37 Flight ID: fltID_28618_4681	Rank: 2 Flight ID: fltID_416129_7488

4.4.2 Individual detector: ClusterAD-Sample

For ClusterAD-Sample, the normal flight operation is also determined but with the GMM as earlier mentioned in section 4.2. The class of a data sample is determined from the type of observed operations or cluster to which it belongs. Figure 4-3 is the graphical representation of the cluster analysis with clusters 0 to 35. Cluster 0-10 and 30-35 highlight in the figure, show the clusters with the most frequently observed

operations based on the probability densities with each cluster representing an operation during that phase. For example, distance zero to touchdown in the figure is the touchdown point and the high density at that point shows that it was a frequent operation carried out by majority of flights.

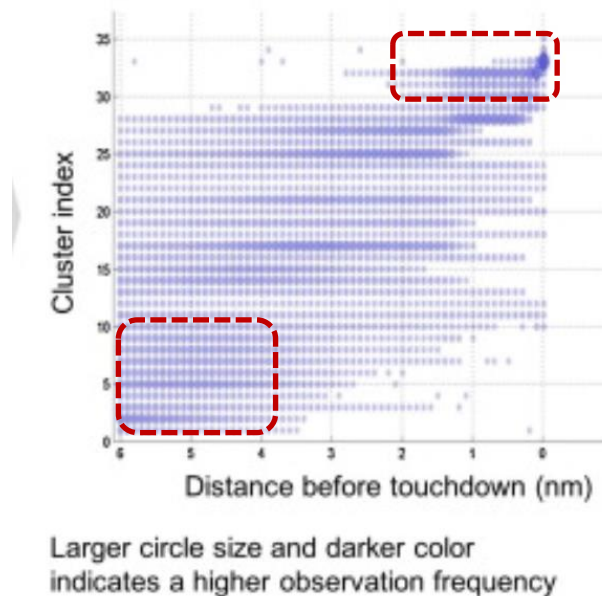


Figure 4-3 ClusterAD-Sample cluster analysis representation [7]

GMM, compared to other clustering methods described in section 4.2, can give statistical inferences of distributions which are in turn used to compute the degree of abnormality of the data samples. Detecting anomalies is the final part of the ClusterAD-Sample process where the probability density function of each sample being normal, is calculated. Mathematically, it is a sum of the pdf of a sample belonging to a cluster, weighted by the pdf of the cluster being appropriate for that flight phase[7].

The general limitation of the ClusterAD approaches is their inability to perform well detecting anomalies in categorical features. One of the core reasons for this is the

heterogeneous data used by this method. This means that both categorical and continuous parameters are normalised using the same scaling technique though they have varying value scales. To address this problem, this study sought to increase the performance of ClusterAD-Sample by segregating continuous and categorical features. The algorithm receives the data subset for each data type independently thereby working on data values with a common scale. The separation process produced variants of the ClusterAD-Sample which are described in section 4.4.3 – 6.

4.4.3 Individual detector: ClusterAD-Continuous (CAD-Cont)

With this variant of ClusterAD-Sample, the 32 continuous parameters are used. The already prepared continuous flight data each with a dimension of 2880 (32 parameters * 90 samples) requires a dimension reduction step. The Piecewise Linear Approximation (PLA) was implemented to decrease the number of points while maintaining the trend of the time series. PLA takes the mean of blocks of consecutive points as shown in Figure 4-4. The figure plots the original 90 samples as well as the PLA reduced profile for the ground speed feature for a flight instance. The profile trends of both plots are similar but with the PLA having a reduced number of samples i.e., 10. For a PLA of 10 (implemented in this work with an input window size of 9), the dimension would be reduced to 320 (32 parameters * 10 samples).

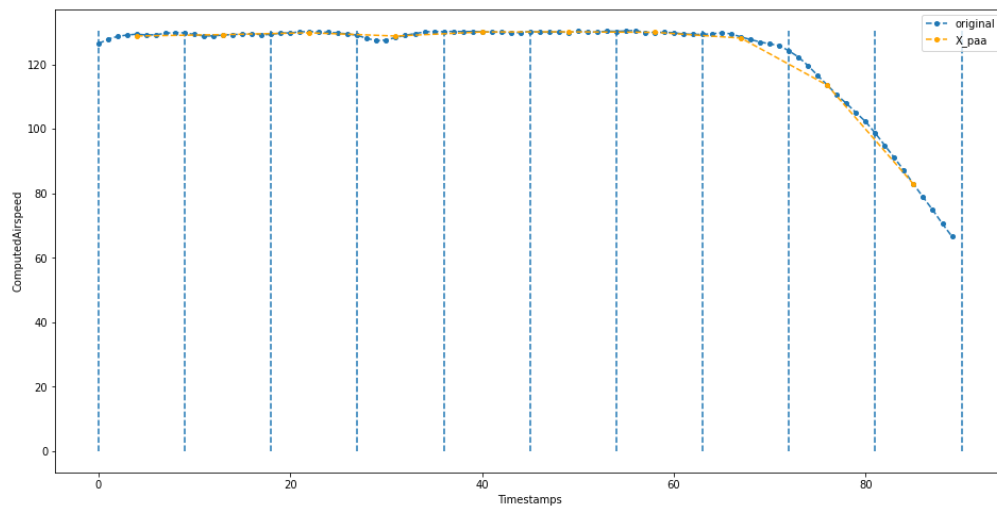


Figure 4-4 Plot containing original and PLA-reduced data for CAD-Cont

CAD-Cont is then implemented and produces an output of likelihood scores [17] for this continuous dataset. The same ROC and CDF metrics were computed giving rise to a threshold estimate of 0.84 and an AUC of 0.9 on training data. This optimal threshold is then applied to new instances of data to ascertain its performance.

No Skill: ROC AUC=0.500
ClusterAD-CONTINUOUS: ROC AUC=0.896
Best Threshold=0.835679, G-Mean=0.847

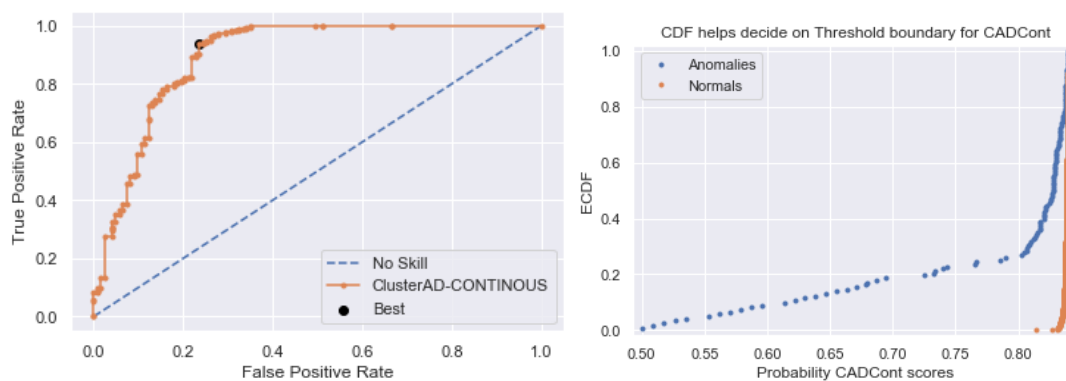


Figure 4-5 (left) ROC and (right) CDF showing CAD-Cont score distribution with optimal threshold

4.4.4 Individual detector: ClusterAD Categorical (CAD-Cat)

The categorical features within the data are fundamentally describing state transitions in landing. Exploration of the data revealed that some state changes are instant, some occur and remain in that state for a while and others simply remain in one state throughout the landing window. It was therefore important to have a representation of the data that gives the coverage of the state indicators in an adequate manner.

Further Dimension Reduction Step

It was deemed useful to have a robust selection of values that describe the categorical values of the data. Although a dimension of 1440 (16 flight parameters * 90 samples) per flight for the separated ClusterAD categorical dataset is better than the entire 4320 (48 parameters * 90 samples), a further reduction in dimension would be even more beneficial to the clustering process. This was carried out by further reducing the categorical data from 1440 to a dimension of 48 (16 parameters * 3 samples). The representation was achieved using three conditions which were determined based on exploratory analysis carried out on the categorical parameters.

The conditions for populating the new CAD-Cat feature space were:

- The time index of the first occurrence of the maximum value. For example, on entering this landing window, the position of the flaps (*flapAngle*) for many flights were set to FLAPFULL ('2' in graphs) throughout while some started with FLAP3 ('1' in graphs) and then switched to FLAPFULL at some point within the 90-second window. A few were FLAP3 throughout. Therefore, for each flight, the maximum feature value (1 or 2) will be observed, and its first-occurring time index inserted as the new value for the *flapAngle* making up the first of the three new samples. Flights that entered the window with FLAPFULL/FLAP3 deployed throughout would put down time 0 as the first occurrence of the FLAPFULL '2' or FLAP3 '1' maximum value respectively. For those flights that entered the

window with FLAP3 deployed and then later switched to FULL, the time index when the pilot deployed the FULLFLAP and that could be a value of 45 as opposed to 0 in the former use case. This ensures that the model can capture exactly when certain actions were carried out within the landing phase enabling a comparison between actions carried out on time and those that were delayed.

- The second criterium observes how the next of the three new time samples for a specific feature. It observes how long the categorical feature remained in the dominant state. This ensures that information about the action a pilot takes is extracted as well as how long it remained in that state.
- The third condition used to reduce the dimension of the categorical feature space is how many times there was a state/mode change for each of them in every flight. This helps reveal events related to unstable state of the aircraft which is often the reason pilots would change state more or less than is required.

Table 4-4 Extract from python output of the new reduced dimension of categorical features

Out[72]:

	AP_1_Engaged	AutoP2Eng	EngagedATS	Flap_Angle	GRight	GNose	GLeft	Brake_OnOffAutomatic	Reverser_Deployed_Eng_1	RevDepldEng2	...
0	0	0	0	0	69	71	69	78	73	74	...
1	0	0	63	90	90	90	90	5	17	16	...
2	1	1	2	1	2	2	2	3	2	2	...

After the reduction process, normalisation of the data could be carried out as the data can now be considered as numeric as shown in Table 4-4. Normalisation is used where there is a need to calculate distances between data points (which is the case in this work). It entails bounding the categorical values between two numbers usually [0,1] or [-1,1]. This step is critical in ML as it makes all features comparable ensuring features with larger magnitude do not dominate the model learning. In this study, the *StandardScaler* in python is used for this purpose and it involves removing the mean

from the value and scales each feature to the unit variance or standard deviation. The function is carried out feature-wise in an independent manner.

A transformation step is then carried out to re-shape the normalised data into the form (F, T, N) where F is the number of the number of flights in the dataset, T is the length of time series vectors (i.e., 3 time series in this case) and N is the feature size (here, 16 features). This transformation prepares the data as input into the ClusterAD model.

The CAD-Cat is the ClusterAD form carried out on only categorical features. It produces the log-likelihood score for each flight and at each time sample e.g. (-224.68316863042872, (207, 2)) with -224.68 being the log-likelihood for flight 207 at time sample 2. The log-likelihood for each flight is determined by averaging the likelihood scores for all the three time samples as shown in the Table 4-5 example below. The next step is to determine the threshold value which helps ascertain which flights are anomalous or not. The CDF and ROC information in Figure 4-6 is a plot of probability scores of all flights from the CAD-Cat model and optimal threshold results.

Table 4-5 Example of CAD-Cat log likelihood averaged scores

	0	1	2	Average
0	-85.035337	-90.951430	-62.055785	-79.347518

The CDF plot reveals that CAD-Cat assigned appropriate scores to the majority of the normal flights except the few in orange which are positioned away from the rest (highlighted in red). These few reveal a lower probability that the flight is a normal flight. The ROC-curve shows the performance of the CAD-Cat model compared to a model with no skill. From the ROC above, the best threshold is 0.83 with a corresponding AUC is 0.87 and when observed from the CDF plot, reveals the reason that threshold is estimated as the optimal since it best distinguishes the model's classification of the flights.

No Skill: ROC AUC=0.500
ClusterAD-DISCRETE: ROC AUC=0.871
Best Threshold=0.838362, G-Mean=0.832

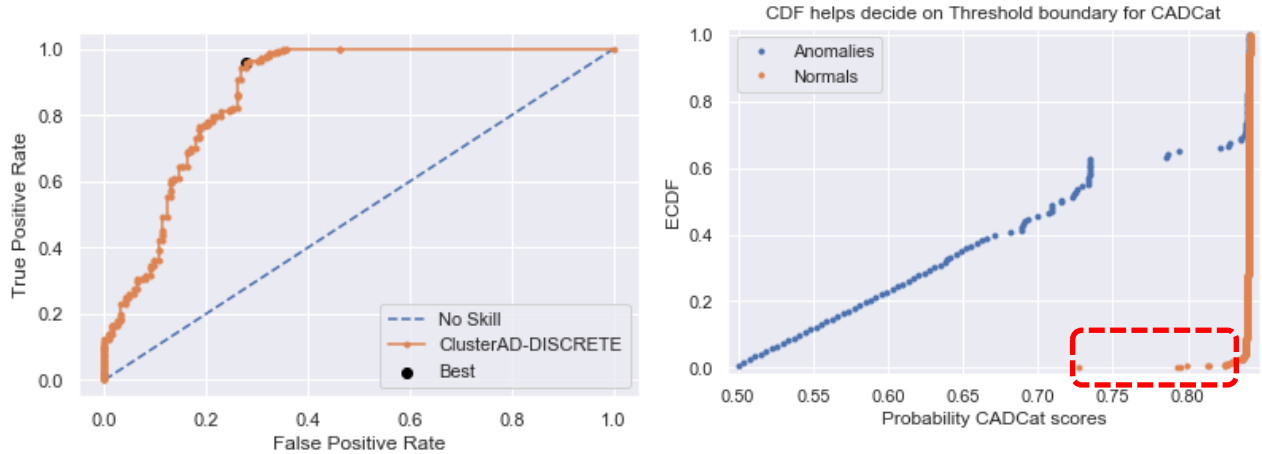


Figure 4-6 (left) ROC and (right) CDF plot information for CAD-Cat also showing the estimated threshold

4.4.5 Individual detector: CAD_ScoreMerge

CAD-Cat and CAD-Cont would be limited in their performance if used individually since the models were implemented on only a subset of features. Therefore, there was a need to merge the outcome of the two models. Here, the resulting scores of CAD-Cat and CAD-Cont were merged to pull the strengths of both models into one. CAD_ScoreMerge final score for each flight was calculated as the lowest probability score of the two i.e., the most anomalous of the two scores. The optimal threshold is determined in the same way as above using an ROC-curve. As shown in Figure 4-7, using a score of 0.897 would result in a higher number of anomalies flagged but also more false positives. A lower threshold means that there would be lower false positives but also a reduction in the anomalous events identified. This work sought to highlight more of these outlier events and so, the G-Mean metric was used to estimate the optimal threshold values. This metric will be explained in the evaluation section.

No Skill: ROC AUC=0.500
ClusterAD: ROC AUC=0.936
Best Threshold=0.834647, G-Mean=0.897

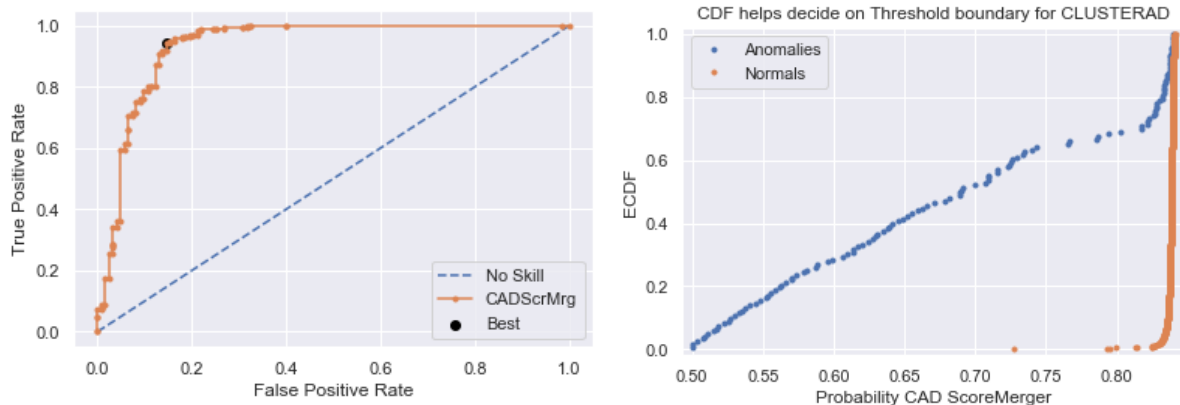


Figure 4-7 (left) ROC and (right) CDF plot for CADScoreMerge

4.4.6 Individual detector: CAD-Aggr

This model combines the labels (in contrast to CAD_ScoreMerger) of CAD-Cat and CAD-Cont using a Logic-OR operation. This means that here, the CAD-Aggr classes a flight as anomalous if either of the two individual models labels it as anomalous. Where both models agree on a label, the outcome of CAD-Aggr shows this agreement. However, if they disagree, this model assumes there is an event of interest and so identifies that flight as anomalous.

4.4.7 Individual detector: Image Processing

'Pure' image processing involves taking an image input and producing a modified image output. Image processing has been employed in several domains and for different applications such as biometric pattern recognition, biomedical image analysis, defence surveillance, image compression and image enhancement [18]. Another line of work is the application of image processing to crack detection in engineering structures [19]. Like anomalous events in flight operations, cracks are

anomalies that need to be detected in time to prevent further issues to the machine and ensure safety. The difference between the crack and flight data is that the former is collected in form of images while flight data exists as sensor measurements. Motivated by the study about crack detection, this current work sought to apply image processing for flight anomaly detection; an area that is yet to be researched in literature. Image processing for flight AD is beneficial as it takes both categorical and continuous data types as input. In this study, the steps involved in image processing for flight anomaly detection are data pre-processing, image creation, similarity matching and anomaly detection.

The algorithm takes as input the heterogeneous dataset from the earlier-described data preparation phase in section 4.3.1. Image creation entails converting the numeric data to image form by transforming the measured values to 0-255 since colours for image pixels are stored as integers that range from 0-255. For each parameter, the maximum and minimum values of the entire data are used for scaling. The resulting data is then stored as images in .png (portable network graphics) format. The similarity matching step imports a set of images for normal flight examples representing the typical normal landing profiles. A similarity measure is then calculated between a query flight instance and the images in the normal cluster using correlation therefore, the technique will henceforth be referred to as Image Correlation (ImageCorr). The output is the corresponding score (between 0 and 1) for each query image depending on how visually similar the image is to those normal images with the least similar match ranked last and having the closest score to 0.

The AD step would use these scores, estimates an optimal threshold as previously described and this decision boundary informs which flight images are dissimilar enough to be labelled anomalous. During training, the method proved very efficient in finding abnormalities with a threshold score of 1 meaning that scores below 1 had anomalies while all normal images had a score of 1 as shown in the ROC of Figure 4-8. Image processing for flight anomaly detection is beneficial. With the image

creation step, some differences in the images can be easily seen with the naked eye while others would be identified by the similarity matching step of the process.

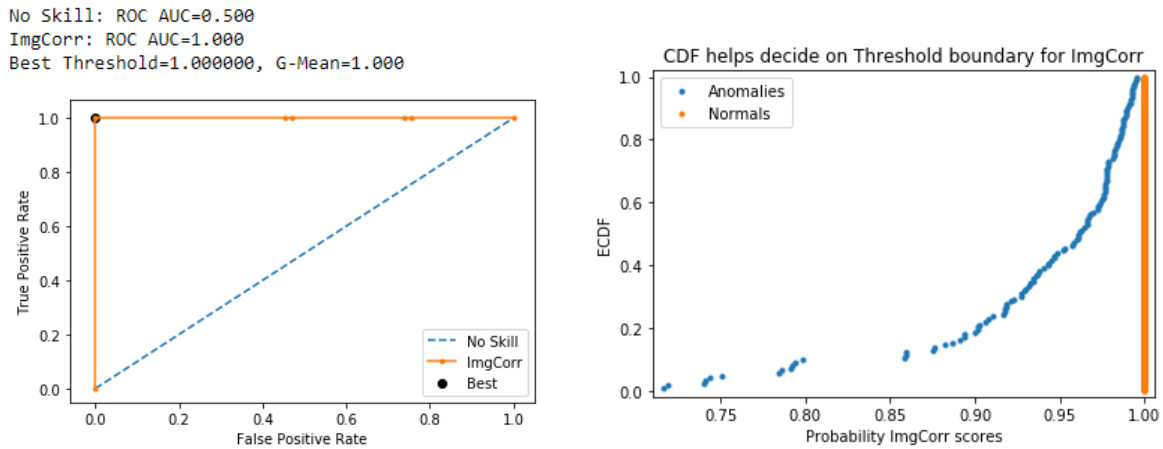


Figure 4-8 (left) ROC and (right) CDF plots for ImageCorr with optimal threshold

4.5 A case for segregating datatypes for CLUSTERAD

CAD-NS (ClusterAD No Split) was the implementation of the ClusterAD-Sample in its original form i.e., with both categorical and continuous features used simultaneously in the model. The statistical metrics used to compare the ClusterAD variants summarised in Table 4-6 below are explained. The results in terms of statistical performance show a higher outcome of 0.94 for CAD-NS; better than other forms (i.e., CAD-Cat, CAD-Cont, etc) of the ClusterAD algorithm described in this work. The main reason for this is its known capability in finding anomalies in continuous data types and continuous features having the highest number of features and in turn, events in this data. CAD-NS however, did not find the highest number of anomalous flights nor did it find the highest number of normal flights as shown in Table 4-7. Its high statistical performance thus, stemmed from its overall performance at correctly classifying both the normal and anomalous flights. CAD-Cat was close behind in performance and

CAD-Cont had the least performance (based on all metrics outlined) for this continuous dataset.

Table 4-6 Table showing the results of all ClusterAD model variants using statistical metrics

Metrics	CAD-NS	CAD-Cat	CAD-Cont	CAD-Aggr	CAD-ScrMrg
Accuracy	0.94	0.93	0.87	0.88	0.91
Precision	0.94	0.93	0.89	0.90	0.92
Recall	0.94	0.93	0.87	0.83	0.91
F1-score	0.94	0.93	0.88	0.84	0.91
G-Means	0.94	0.93	0.88	0.86	0.91

Table 4-7 Table showing the results of all ClusterAD model variants using populations.

TP	85	74	81	90	87
TN	278	286	257	250	265
FN	8	19	12	3	6
FP	17	9	38	45	30

From a coverage or contextual standpoint, however, CAD-Aggr identified more anomalous events correctly i.e., TP thereby reducing significantly, the FNs flagged although relatively producing a higher number of FPs. Specifically, it was discovered that CAD-Aggr was able to find every anomaly detected by CAD-NS and more. An example of the additional anomalies flagged by CAD-Aggr was a use case (flight

65358) where the *NoseGear* was held off the ground at TD for longer than usual. Figure 4-9 shows the left, right and nose gear profiles of the landing window for this anomalous flight compared to that of a normal flight. The top figure with the anomalous event shows that the gear switches were all in ‘Gear-down-air’ mode on entering the 90-second window. However, while the left and right gear switches were transitioned to ‘Gear-down-ground’ mode at touchdown, the nose gear switch to ground mode was delayed for at least 6 seconds after the other two gear modes changed. Judging from the normal profile in the bottom figure, this event deviated from the expected norm. An expert provides more context to the significance of this flight:

“.. it is a deviation, albeit a low risk one. If I was running the FDM programme in the airline and saw this, it would prompt me to look at the pilot’s other flights to establish whether there is a technique issue. If it’s a technique issue, that’s really worth knowing about.”

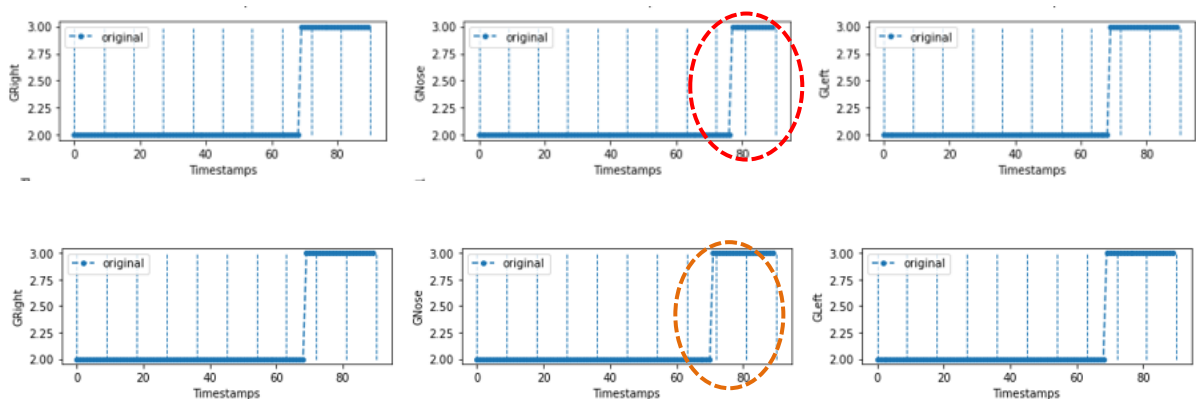


Figure 4-9 (top) Profile of GearRight, GearLeft and anomalous event in GearNose; (bottom) Normal profile of GearRight, GearLeft and GearNose

This case was identified only by CAD-Cat thereby demonstrating that the reduced dimension with the categorical variables is a good representation of the original categorical dataset which in turn highlights the capability of CAD-Cat. It is believed

that CAD-Cat did well in flagging this type of anomaly because it consists of data regarding the existence of a state (or action), the time the action occurred and how long it stayed in that state. Therefore, this use case explains its ability to identify delays in actions (as is the case in this scenario) or steps carried out earlier than expected. The lower number of flights produced by CAD-Cat and CAD-Cont were because of the reduced subset of features used while implementing them as opposed to CAD-NS which made use of the entire feature set. From the exploration, it was evident that more anomalies were present in continuous parameters than the categorical features. Therefore, it was to be expected that CAD-Cont found more anomalies than CAD-Cat.

Among the ClusterAD forms, the results of CAD-Aggr was preferred despite its high FPs. This choice is based on the premise that providing additional domain safety knowledge whilst reducing FNs (missed anomalies) are important factors in FDM today. In other words, despite the high statistical result for CAD-NS, this study chose to carry-on the label outcomes of the CAD-Aggr unto the ensemble AD because it covered more ground in identifying wider array of anomalies than the other ClusterAD forms. This provides evidence on the benefit of segregating data types prior to model learning and then re-combining their outcomes afterwards as carried out with CAD-Aggr. This outcome provided confidence in the decision to recombine these ClusterAD variants in the proposed ensemble AD framework.

4.6 Evaluation: Individual Models

This current study evaluates the individual models by using their performance at classifying all flight instances as opposed to selecting only the top 1%, 3%, 5% or 10% of the anomalies found as is the case in previous literature [8]. This allows for measuring the performance of these methods by using several statistical metrics such as accuracy, geometric-mean as well as a confusion matrix measure which exposes how they perform overall on the entire flight population. Also, evaluating using all flight instances ensures every anomalous event, identified in every flight by the models, is

brought to light for further investigation as they may be precursors/triggers to accidents. For example, the use case previously described with the NoseGear delay, is an unusual event that qualifies as an anomaly even though the expert regards it as 'low risk' deviation from the norm. However, this type of risk would prompt further investigation by an analyst to establish whether there is a technique issue by the pilot or if there is a deviation normalisation pattern among a significant number of pilots which need to be corrected before it leads to a severe occurrence.

Detecting more outlier events in flight operations provides extra safety knowledge and in turn leads to tighter mitigation steps which would ultimately contribute to enhancement of safety in the industry. The evaluation is carried out using both statistical measures as well as examining their contextual performance that reveal their unique AD characteristics.

4.6.1 Statistical Analysis of individual methods

In applied Machine Learning, comparing different methods with a view to selecting a final model with the best outcome, is quite common. In doing this, studies focus on measuring their performance using statistical metrics. The metrics considered in this work are accuracy, precision, recall and G-Means. Confusion matrix is used to derive some of these metrics.

The **confusion matrix** comprises of rows representing the instances in an actual class and columns representing the instances in a predicted class. Table 4-8 below describes the insights the matrix provides in terms of the population of instances that were correctly and incorrectly classified by the detector. An ideal matrix would have no FP and FN since they are the population that it incorrectly classed. This is however not often the case, hence the need to develop models that reduce these incorrect predictions. There is often a trade-off between missing anomalies and false alarms. Since this work is aimed at enhancing flight safety, stakeholders prefer a method that

can produce the least number of missed anomalies over higher numbers of false alarms.

Table 4-8 Table showing confusion matrix description

Actual/Target label	Detector Predicted label			
		Anomaly (P)	Normal (N)	
	Anomaly (P)	Correctly predicted anomaly (TP)	Incorrectly predicted anomaly (FN) i.e., missed anomaly	Total Actual Anomaly instances in data
	Normal (N)	Incorrectly predicted normal (FP) i.e., false alarm	Correctly predicted normal (TN)	Total Actual Normal instances in data

Precision seeks to answer the question: Out of the anomalous instances predicted by the detector, how many instances were correctly classed? In other words, precision measures how specific a detector is at identifying the anomalous instances that is, $TP/TP+FP$.

On the other hand, **Recall** is a metric that measures how sensitive a detector is at highlighting correctly classed anomalies out of the actual number of anomalies present within the data. It answers the question: Among the anomalous instances in the entire dataset, how many instances were correctly classed by the detector? It is calculated using $TP/TP+FN$. Considering the aim of this work is to find as many interesting anomalous cases as possible, this metric is a useful one to implement for comparing the detector outcomes.

```

ImageCorr Anomalies
[[116 10]
 [ 16 226]]
Accuracy Score : 0.9293478260869565
Report :

```

	precision	recall	f1-score	support
-1	0.88	0.92	0.90	126
0	0.96	0.93	0.95	242
accuracy			0.93	368
macro avg	0.92	0.93	0.92	368
weighted avg	0.93	0.93	0.93	368

Figure 4-10 An example showing Statistical metrics described

The **F1-score** combines the precision and recall of a classifier into a single metric by taking their harmonic mean. This metric is expected to give a balance between how specific a detector is to its ability to be sensitive to anomalies. Accuracy computes how many predictions (both normal and abnormal classes) were correct from the total i.e. $TP+TN/TP+TN+FN+FP$. Figure 4-10 above provides an example of computed results that these metrics return.

Since there are often more normal than abnormal instances, a detector could be 99% accurate but not be able to find any abnormal cases in a flight dataset. Therefore, there is a need for a better measure. For this research, the accuracy metric was not beneficial. Geometric mean (G-Mean) was however implemented to meet this need. Computing the accuracy for positives and negatives separately, to measure how well the detector performs for one and the other is useful here. In other words, a “positive accuracy” ($TP/TP+FN$) and a “negative accuracy” ($TN/TN+FP$) also known as **sensitivity and specificity** is introduced. G-Means combines these two indicators into a single value by finding the square root of their product. Like F1-score, G-Means provides a useful property of averaging out these measures while penalising their unbalanced pairs.

Table 4-9 showing the performance of individual detectors implemented with green cells highlighting the best performing models based on each metric. The metrics produced similar values and so, this aspect will be focusing on the G-Means metric in subsequent evaluations in this study.

Table 4-9 Summary of statistical performance of the individual detectors

	MKAD	CAD-Cat	CAD-Cont	Image Corr
Accuracy	0.89	0.93	0.87	0.93
Precision	0.89	0.93	0.89	0.93
Recall	0.89	0.93	0.87	0.93
F1-score	0.89	0.93	0.88	0.93
G-Means	0.89	0.93	0.88	0.93

Table 4-10 Summary of classification populations for the individual detectors

TP	76	74	81	79
TN	268	286	257	281
FN	17	19	12	14
FP	27	9	38	14

As previously discussed, CAD-NS produced the highest G-Mean of 0.94 based on the number of correct classifications it identified. It however missed some anomalous events which CAD-Cat was able to highlight. In terms of statistical G-Mean, CAD-Cat

and Image Correlation performed second best with 0.93. Table 4-10 above also reveals that despite CAD-Cat finding the lowest number of anomalies compared to other methods, it had the highest number of correctly classed normal instances; expected due to the low anomalous events it was able to flag. Due to CAD-Aggr's high performance in correctly detecting anomalies, there were fewer anomalous events missed by this model.

The performance of these individual models confirms the findings in[8] that ClusterAD outperforms MKAD, which is the baseline algorithm, in detecting significant anomalies. The authors of the above-mentioned study also concluded that MKAD performed better in detecting anomalies present in categorical features while ClusterAD did better identifying anomalous events in continuous datasets. This conclusion was also reached in this current work as described in the next section which deals with the Contextual performance of these individual models. This contextual evaluation of the individual models was essential as it (alongside the statistical performance) drives the choice of models that make up the ensemble AD.

4.6.2 Contextual Analysis of Individual Methods

In this evaluation method, the abnormal flights identified by the different models were compared to ascertain the common and the differentiating characteristics of each. The review of an expert pilot and flight analyst is essential here, to gain operationally beneficial insights from the abnormal flights identified. In other words, the expert review helped validate the abnormal events found as well as confirm procedure deviation anomaly types. The baseline used in this work is MKAD. The detectors were implemented on the same dataset from an international airline described in section 4.3.2. Flights detected could only be validated by the authorised expert with access to the data so as not to breach data protection protocols. Table 4-11 summarises the

comparison tests carried out which also aided the decision of detectors to use for the ensemble.

Table 4-11 Table summarising the evaluation tests implemented.

Test cases	CAD-Cont	CAD-Cat	ImageCorr	Ensemble	Evaluation Technique
<u>Evaluation I:</u>					
Common Flights detected by all detectors	✓	✓	✓		Comparison with MKAD
<u>Evaluation II:</u>					
CAD-Cont, MKAD	✓				Comparison with MKAD
CAD-Cat, MKAD		✓			
ImageCorr, MKAD			✓		
<u>Evaluation III:</u>					
Ensembles LOGIC-OR, WM, MV, WJ				✓	Comparison between Ensemble methods
<u>Evaluation IV:</u>					
Best Ensemble, MKAD				✓	Comparison between Ensemble with MKAD

Evaluation Study I: Common Flights Detected by All Models

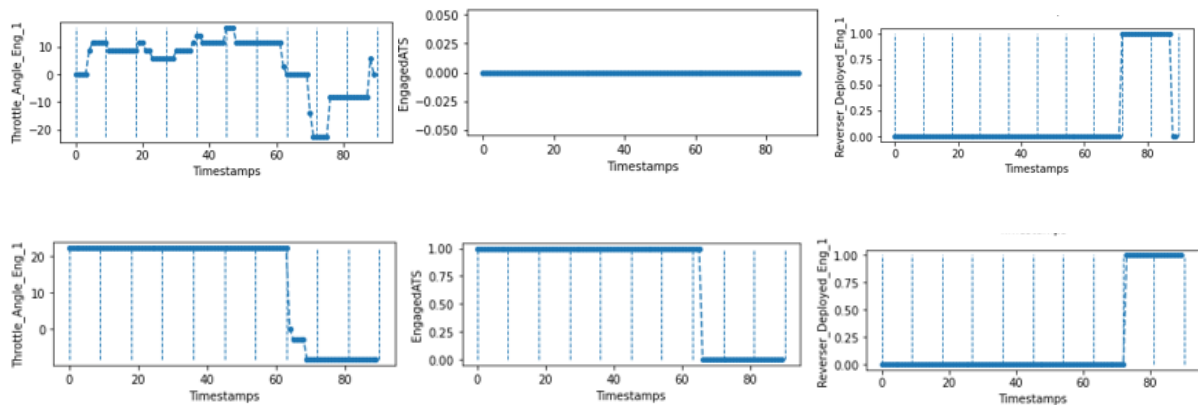


Figure 4-11 (top) shows feature profiles of anomalous flight 249714. Figure (bottom) shows normal profile for selected features

Figure 4-11 compares the normal (bottom) feature profile to a flight (top) with an unstable final approach as well as a possible deep landing. Anomalies were found in both continuous and categorical features for this flight. This use case exposes an unusual technique where the ATS (Auto Thrust System) was inactive from the final approach and remained this way throughout landing. This is unexpected since the ideal procedure is for the ATS to be active in the final approach but only disengaged just before touchdown. At touchdown, the normal profile shows the throttle lever angle for one of the engines reducing to zero meaning that the throttle levers have been retarded to IDLE mode. The thrust levers help control the power output of the aircraft engines and their current angles describe the position of the lever. In other words, the greater the lever angle, the greater the thrusts to the engine. For the anomalous flight, the figure reveals lower and unusual fluctuations to the throttle angle which combined with a completely disengaged ATS, informs me that the flight must have been flying at reduced power than normal from final approach leading to the unstable state. All the ClusterAD detector variants identified this flight as anomalous due to the fluctuation in the *throttleAngle* continuous parameter. MKAD flagged this flight because of an extra

step in the sequence of crew actions relating to disengaging the engine reversers in the landing roll. This was unusual and the action is deemed too early since majority of flights did not disengage in this 90-second window. Although this action was not an anomaly, it deviated from the expected normal profile drawn from most flights.

Evaluation Study II: Flights detected by individual models compared to MKAD

Comparison between CAD-Cont and MKAD

This use case describes distinguishing characteristics of the ClusterAD and MKAD models. The comparison between CAD-Cont and MKAD is described with three use cases that represent anomalies that both MKAD and CAD-Cont detected, those that MKAD did not find but CAD-Cont did and those that MKAD identified but CAD-Cont did not.

Use case 1: CAD-Cont = yes; MKAD = no

The flight in Figure 4-12 provides evidence that confirms MKAD is less effective in finding flights with anomalies only in continuous parameters. In the figure, anomalies were found in the roll side stick for the FO (First Officer) particularly because inputs from that sidestick was unexpected given there were already inputs from the captain at the same time. These double inputs already constituted a deviation from standard operating procedure. There was also a low pitch attitude revealed in the pitch angle profile in Figure 4-13 below. Although the windspeed as revealed in Figure 4-14 shows unusually high winds possibly leading to a turbulent approach, the other inputs from the flight crew shows some interesting choices worth investigating further. MKAD was unable to find this flight possibly because all abnormalities were not only in continuous features but like the low pitch attitude, the anomalies were not prolonged but instantaneous.

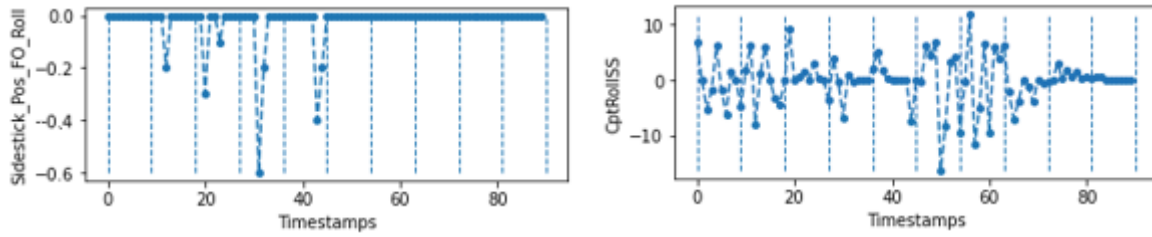


Figure 4-12 Plot of abnormal flight 153633 with double inputs from both FO and Captain

Figure 4-13 gives visual evidence of the pitch angle profile that suggests a low pitch attitude late on in this window. Also Figure 4-14 confirms a higher windspeed than normal for this abnormal flight.

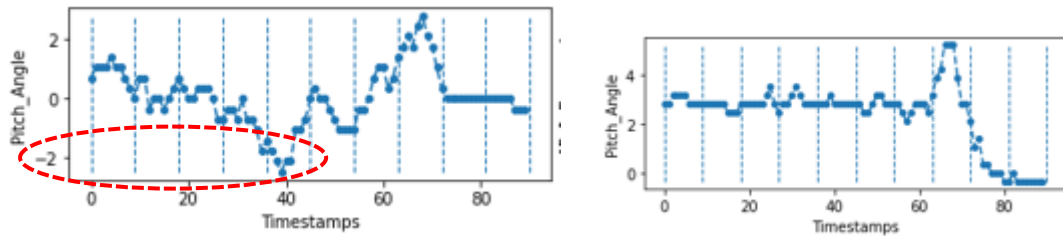


Figure 4-13 (left) Plot of Pitch Angle in abnormal state. (right) Pitch Angle in normal state

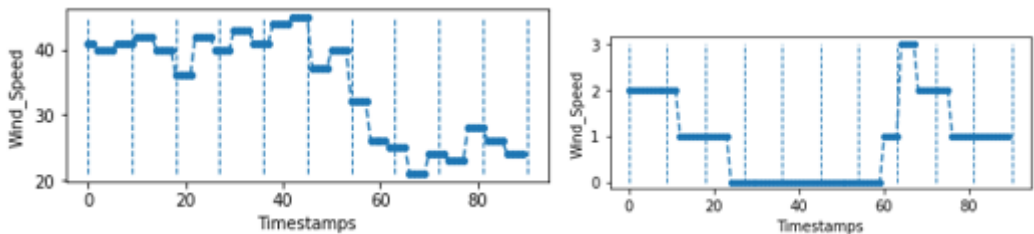


Figure 4-14 (left) High Windspeed (right) Normal windspeed

Use case 2: CAD-Cont and MKAD = yes

Evidenced in Figure 4-15 is flight 416129 with its slat position at different points in the landing window with reference to the touchdown point, a perspective the altitude plot

provides insight about. Both detectors found the anomaly in this use case. MKAD and CAD-Cont both detected that the flight landed with lower-than-normal slat position of 22 degrees as opposed to 27 degrees used by the majority. Slats are devices that run along the edges of the aircraft wings used to decrease stall speed during important manoeuvres like landing. For the A320 aircraft used in this study, the slats and flaps are controlled by the same lever annotated as ‘Flaps’ and there are five different positions on that lever. The last position of the lever is flapFULL when the slats move from 22 to 27 degrees and this was the lever position most of the flights were in, for landing. This use case, however, had the lever at flap 3 (4th lever position) for the landing hence the slats were at 22 deg. This is not necessarily an incorrect action but is different from the majority that landed with slats in the flapFULL position. There could be reasons like weather which prevents the pilot from landing with the levers in the flapFULL position but nevertheless, such flights make the list of interesting flights that an analyst would want to investigate.

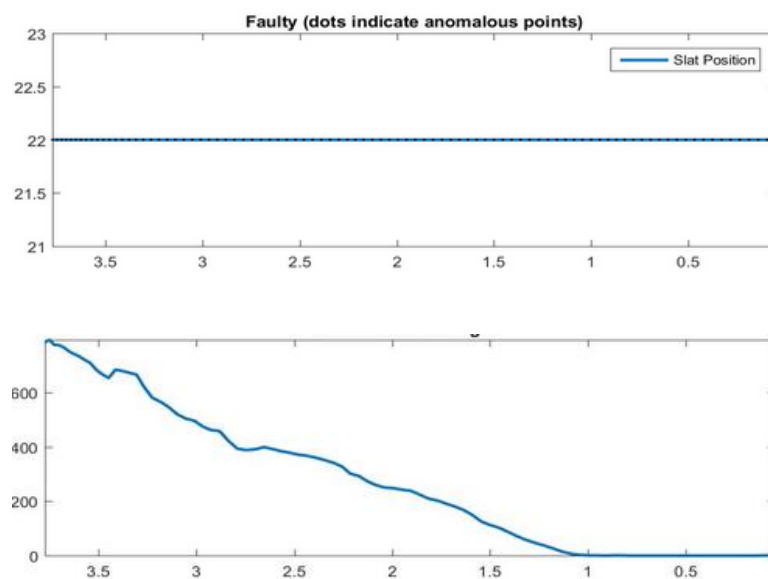


Figure 4-15 Plot of Slat position and Altitude for a flight of interest

The below Figure 4-16 highlights actions (solely in categorical variables) carried out by the crew as well as the sequence in which they occurred. Therefore, in addition to

the unusual slat position, which is an event in a continuous parameter, MKAD also found some abnormalities in the sequence of actions of the categorical features. Actions in blue indicate any extra action that was carried out which is deemed different from the normal profile. The actions in red symbolise actions that are missing at that point in the sequence of actions. The ‘flap_Angle_On’ feature in black describes the flap lever was transitioned into the normal/expected position (i.e., flap3) at the start of the window. There was then an expectation that it would be further changed to flapFULL, but this did not happen for this flight as shown by the red ‘Flap_Angle On’ action in the figure. The figure also depicts that this flight had 72% of anomalies in discrete parameters than the 28% in the continuous parameters.

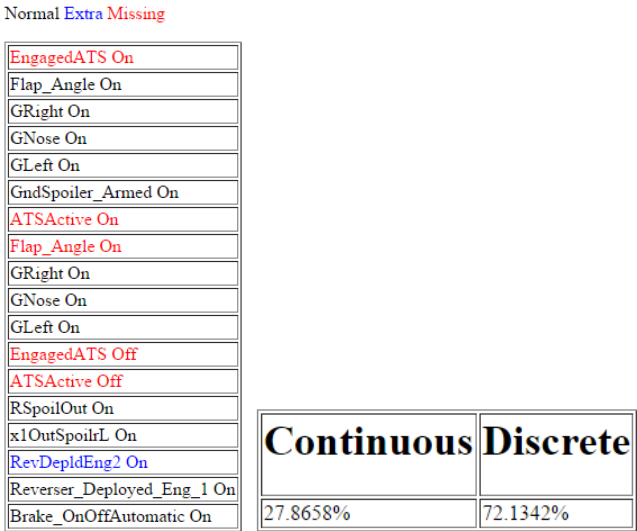


Figure 4-16 MKAD output providing insight into unusual sequence of actions

Use case 3: MKAD finds continuous anomaly

This use case represents a set of flight anomalies that exist in continuous parameters which MKAD is able to detect. It confirms that MKAD does find a few anomalies present in continuous parameters. In this use case, it is an obvious difference in roll

angle and sidestick positions compared to the normal profile. Also, Figure 4-17 reveals (using the black dots on the lines) the many different points during the flight profile which the detector classes as a deviation from the norm. The fact that there are many black dots also prove that MKAD finds some anomalies in continuous variables since it means the erroneous behaviour lingers or the profile shows several instantaneous deviations that add up to help flag the occurrence.

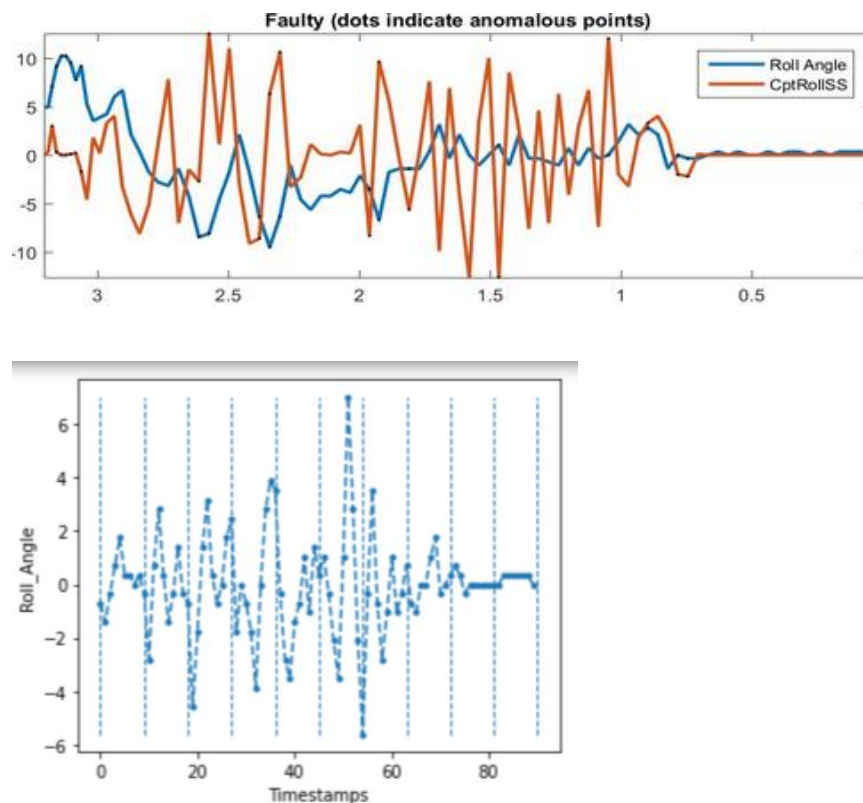


Figure 4-17 (top) Abnormal Roll angle and position profile (bottom) Roll Angle Profile of a normal flight

The Figure 4-18 below shows that AP1 for the same flight was engaged and then disengaged just as it entered the landing window. It was expected that the landing configuration would have been set prior to the start of this window but instead, AP1 was disengaged in the first few seconds. This occurrence, in addition to the roll angle anomaly, indicate a likely unstable final approach for this use case. The use case

shows that MKAD did not find the single instantaneous anomaly even though it was present within the categorical Autopilot features.

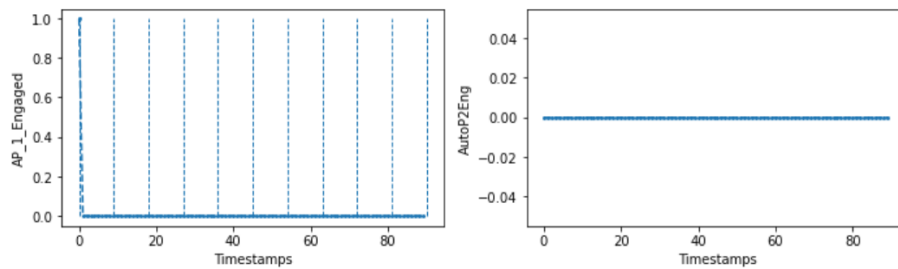


Figure 4-18 Instantaneous anomaly in both Autopilot features

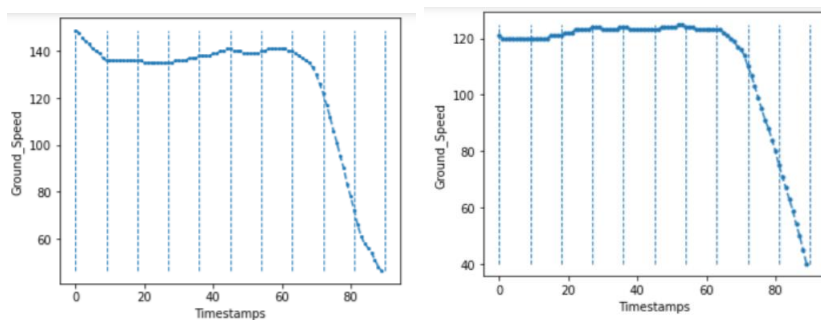


Figure 4-19 (left) High ground speed profile (right) normal ground speed profile for majority of flights

Also, in this flight, MKAD was unable to detect the high-speed anomaly in groundspeed profile. Figure 4-19 reveals the abnormal groundspeed profile as well as that of the normal profile in comparison. This use case in general demonstrates an unstable approach which ideally should have warranted a go-around, but the flight crew chose to land despite this. Below Figure 4-20 evidences the requirements in final approach.

FINAL APPROACH

The speed trend arrow and FPV help the flight crew to make timely and correct thrust settings (if in manual thrust) and approach path corrections. Avoid descending through the correct approach path with idle thrust. (Late recognition of this situation without a prompt thrust increase may lead to considerable speed decay and altitude loss).

• At 1,000 FT AAL

- **STABLE APPROACH CRITERIA** **CHECK**
Refer to SOP 3.03.01 'Stable Approach Policy' for guidance on the required action.

Note: In gusting winds the Vapp speed target is likely to change rapidly and/or frequently due to the function of GS mini. In these circumstances, short term exceedance of Vapp by more than 15 kts may be accepted, provided only small corrections are necessary to bring the aircraft back to a stable condition.

• At 500 FT RADIO HEIGHT

- **PNF ANNOUNCE** **"FIVE HUNDRED RADIO"**
Alternatively PNF may announce "RADIO" after the "500" Rad Alt auto call.

- **PF ANNOUNCE** **"STABLE" or "UNSTABLE – GO AROUND – FLAPS"**
Depending on whether stable approach criteria have been met.

*Note: In the event of a go-around with FDs OFF, the FDs will automatically reappear.
In SRS/GA TRK mode, when TOGA thrust is applied.*

WARNING

ACHIEVING STABLE APPROACH CRITERIA IS A MANDATORY REQUIREMENT FOR EVERY APPROACH.

- Avoid any tendency to "duck under" in the late stages of the approach.
- Avoid destabilising the approach in the last 100' to give the best chance of making a good touch down at the desired position.

Figure 4-20 Extract of Standard Procedure for final approach available to crew of the airline

Use case 4: 372388 MKAD=yes. CAD-Cont/CAD-NS= no

The flight profile in Figure 4-21 below describes a use case where CAD-Cont did not detect the event because it manifested as an anomaly solely in categorical features. MKAD on the other hand, was able to detect the anomaly. The figure shows autopilot2 entering the window while in the engaged state and then was disengaged nearly 10 seconds after and stayed this way for the rest of the flight indicating a manual landing. Also, this flight could be said to deviate from standard operating procedures since both autopilots were in different states for the 10 second duration.

AutoP2Eng On
EngagedATS On
Flap_Angle On
GRight On
GNose On
GLeft On
GndSpoiler_Armed On
ATSActive On
Flap_Angle On
GRight On
GNose On
GLeft On
AutoP2Eng Off
EngagedATS Off
ATSActive Off
RSpoilOut On
x1OutSpoilrL On
RevDepldEng2 On
Reverser_Deployed_Eng_1 On
RevDepldEng2 On
Brake_OnOffAutomatic On
Brake_OnOffAutomatic Off

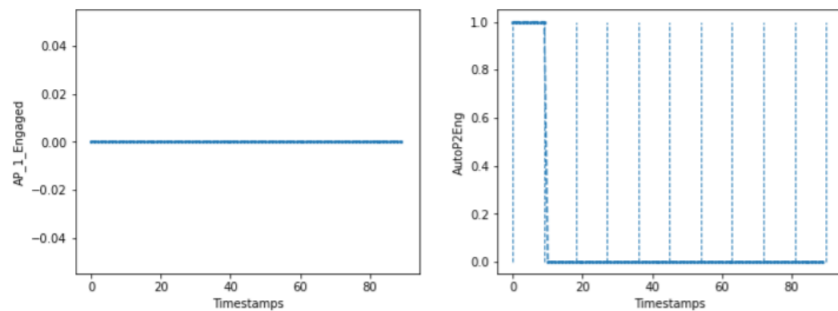


Figure 4-21 Anomalous occurrence in sequence of action and autopilot modes

Comparison between CAD-Cat and MKAD

From second use case in Evaluation II, it is confirmed that MKAD performs better at identifying outliers in categorical parameters than in continuous values. With CAD-Cat also trained on just categorical values, it is expected that both detectors would be able to detect anomalies in categorical data. However, it was observed that CAD-Cat was able to find some anomalous events in categorical variables that MKAD was not able to highlight. An example of this is the flight described previously in Figure 4-9.

The difference is likely due to MKAD creating a sequence representation of actions in the time window using Sequence Miner and SAX approximations and thus can flag an abnormal sequence of actions. However, in this use case, the NoseGear deploy did occur in the appropriate position in the sequence of actions but there was a delay in

making the transition compared to the normal profile for the same parameter. It was this delay that MKAD was unable to identify but which CAD-Cat highlighted accurately. This characteristic of CAD-Cat contributes to the high coverage performance of CAD-Aggr.

Comparison between Image Correlation and MKAD

As described in section 4.4.7, the image correlation detector creates an image representation of the query instance. It calculates the correlation between the query and the normal instances, then decides if that query instance has a high similarity with any of the normal input examples for it to be classed as normal or a low match leading to an anomalous label. Since image correlation used the heterogeneous data as input, it was able to identify anomalies in both categorical and continuous data types. Figure 4-22 provides an example of where the ImageCorr detector was able to find anomalies in different datatypes. The sections highlighted in red, are those with visible differences from the normal images presented in the figure. It is interesting to note that the more visual differences noticed in the image, the more the anomalous features within the flight. For example, with flight 249714, there are 3 areas showing differences from the normal image profile. However, flight 65358 in Figure 4-23 presents a single section of the image with a visual difference (compared to normal profile shown in Figure 4-24) possibly corresponding to the NoseGear feature anomaly for that instance.

During training, Image Correlation found all the anomalous occurrences, better than MKAD and other detectors. On a new set of instances, it was therefore, expected to continue to exceed their performance. However, the model missed out some anomalies. Nevertheless, it performed better than MKAD in detecting certain anomaly types.

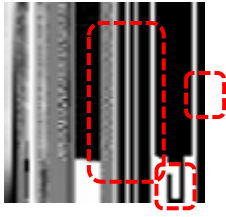


Figure 4-22 Flight 249714 with anomalous event found by ImageCorr for events detected in throttleAngle, ATS(Autothrust) and engReverser

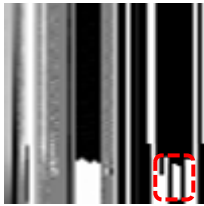


Figure 4-23 Anomaly not identified by ImageCorr due to event only in NoseGear feature



Figure 4-24 Flights 105815 and 106201 respectively are normal flight profiles

4.7 Ensemble Techniques

Several techniques were implemented in a bid to determine an appropriate combination technique for merging the individual detectors. These ensemble techniques are Logic-OR, Weighting by Judgement (WJ), Majority Voting (MV), Weighted Majority (WM). For this work, it was decided that the input into the ensemble would be the predicted class labels from the individual detectors. The individual detectors that were used in this ensemble framework were MKAD, CAD-Cont, CAD-Cat and Image Correlation. The output of the ensemble is the final class prediction for

the flights in the available dataset as well as insights into the abnormalities that caused the anomalous flights to be flagged.

4.7.1 Logic-OR

The Logic-OR is a binary operator that requires two operands as input. This technique will return true if any of the operands or conditions evaluates to true. In this work, the Logic-OR ensemble technique classes a flight as abnormal if any of the detectors also predicts it as anomalous.

4.7.2 Majority Voting

In this technique, the prediction arrived at by majority of the detectors becomes the final output of the ensemble. For example, if MKAD, CAD-Cat and Image Processing predict a flight as normal and CAD-Cont predicts the same flight as abnormal then, the flight would be considered as belonging to the normal class. This would be the final prediction based on the votes of the detectors. Where an even number of detectors is used in an ensemble and the votes tie, this work classes it as a flight of interest to investigate.

4.7.3 Weighting by Judgement

This technique assumes that a model which generally performs better should be trusted more than the others. Therefore, rather than giving each model equal weights, this technique makes use of the performance of the detectors and makes a judgement call on what weights to assign to each model. For this work, the Geometric Mean G-Mean () value for the models is used to decide on the general performance of the detectors, the metric informs the weighting and all weights assigned to the models sum to 1. The weights are multiplied to the individual predictions and the results summed. The result should be a number between the two label values, in this case, between -1 and 0 (with -1 being the anomalous class). In this study, the technique is

provided with a further condition that for each flight where the weighted sum is greater than -0.5, then that flight is classed as normal since it tends towards 0 (i.e., the normal class). However, a weighted sum less than -0.5 for any flight means that it tends to -1 and hence belongs to the anomalous class.

4.7.4 Weighted Majority

This algorithm [20] developed by Tom Mitchell which employs the use of learned weights, makes its final predictions based on the weighted vote among the individual detectors. It learns by changing the weight associated with each model in each sample iteration. The algorithm begins by assigning a weight of 1 to each detector while considering the training examples such that whenever a model misclassifies a new training example, its weight is decreased by multiplying it by some number β , where $0 \leq \beta < 1$. In this work, a β value of 0.9 was used after several iterations during training. This technique was considered for this work because of some properties it possesses. One of the characteristics of the algorithm is its ability to cater for inconsistent training data since it reduces the weight of an inconsistent hypothesis as opposed to eliminating it.

4.8 Evaluation: Ensemble AD

Having analysed the performance of the individual detectors, it was evident that a combination of CAD-Cont, CAD-Cat, MKAD and ImageCorr would produce a better outcome than the baseline detector, MKAD. The decision was based on the unique characteristics or strengths discovered in the individual detectors as discussed in section 4.7. In summary, ImageCorr performs well in detecting anomalies in both continuous and categorical variables and can be beneficial where a flight can be classed simply based on visual inspection. Although MKAD is also able to find abnormalities in both data types, it excels best in finding categorical features where a sequence of actions differs from the normal profile. This confirms the same conclusion reached in literature. CAD-Cont, on the other hand, performs well in identifying

anomalies in continuous parameters. Being tailored to categorical data alone, CAD-Cat is also able to detect significant anomalies in categorical features and is faster than MKAD. This detector is however unique in its ability to detect delayed actions or actions performed earlier than normal. Combining these unique characteristics into an ensemble ensures that the strengths of the individual detectors are integrated thereby producing an outcome that can identify a wider range of anomalies as well as a higher confidence in the outcome. The result of the ensemble techniques applied on a subset of data are summarised in section 4.8.1. The preferred ensemble anomaly detection technique is then compared to the baseline MKAD.

Evaluation Study III: Comparison of Ensemble Techniques

Table 4-12 below summarises the statistical performance of the different ensemble techniques. The three metrics G-Mean, recall and F1-score all agree that WJ method and Logic-OR produce the highest and lowest performance respectively of all the techniques considered. The confusion matrix components (TP, TN, FN, FP) however, reveal Logic-OR as finding the greatest number of anomalies although also having the highest proportion of FPs (false alarms). Close behind are the MV and WJ technique with less population of false alarms but a considerable number of missed anomalies compared to Logic-OR.

Logic-OR detected all flights that other techniques were able to detect and more. This is due to its ability to flag any flight as abnormal when at least one individual detector detects an anomaly. This is unlike the MV ensemble, for instance, which fails to identify anomalies because the algorithm is designed to go with the ‘opinion’ of most detectors in its ensemble. Logic-OR effectively combines the unique strengths of each detector at finding anomalies compared to other ensemble methods.

Table 4-12 Summary of statistical performance of Ensemble using the different combination techniques

	Logic-OR	Majority Voting (MV)	Weighted Majority (WM)	Weighted by Judgment (WJ)
G-Mean	0.88	0.93	0.93	0.94
Recall	0.86	0.93	0.93	0.94
F1-score	0.86	0.93	0.93	0.94
TP	123	118	116	118
TN	192	224	226	227
FN	3	8	10	8
FP	50	18	16	15

Evaluation IV: Comparison of Best Ensemble with MKAD

The Logic-OR combination technique produced a final outcome that surpassed the performance of MKAD in terms of the variability of anomalies it can detect. In terms of G-Mean metric, MKAD outperformed the chosen ensemble with a score of 0.89 compared to 0.87 by Logic-OR. On a dataset with new instances, the ensemble detected 14 more anomalies than MKAD and missed only 3 anomalies compared to 17 undetected by MKAD. The results in Table 4-13 demonstrate that the ensemble can detect anomalies that manifest in continuous and categorical features especially those present in sequence of crew actions. The ensemble has also shown improvement in detecting unusual abnormals that previous methods were unable to find i.e. actions which, although, followed correct sequence but were slightly delayed/early in carrying it out. This study demonstrates that an ensemble anomaly

detection provides more benefit than the current state-of-the-art anomaly detection method.

Table 4-13 Comparison of Final Ensemble performance with MKAD

	MKAD	Ensemble: LOGIC-OR
G-Means	0.89	0.87
Recall	0.89	0.87
Fi-score	0.89	0.87
TP	76	90
TN	268	233
FN	17	3
FP	27	62

A Probabilistic Ensemble method was the gap in this current research as it was not implemented due to time constraints. Probabilities summarise the likelihood of an event as a numerical value between 0 and 1 and is often assigned for each class by machine learning models. However, for the individual detectors considered in this work, their outputs were single scores showing the likelihood of each flight instance belonging to the normal class. Therefore, it is perceived that the Probability ensemble would produce similar results to the Weighted by Judgement technique.

4.9 McNemar (chi-squared) Statistical Comparison: MKAD and Logic-OR Ensemble

This section describes a statistical hypothesis testing method that is applied to classification models while also comparing these algorithms. The statistical testing

interprets ML results by evaluating the models using approaches for algorithm comparison and is based on the target classification of the dataset. There are several statistical hypothesis frameworks applied to difference of proportions tests but for comparing classification models, McNemar's test is introduced in this work. This is because the McNemar's test is a common statistical analysis technique used for classification results and since the comparison of two models is involved here, it was deemed suitable.

In this section, the statistical performance of the proposed Logic-OR ensemble (see section 4.7.1) is explored using the McNemar test which also allows its performance comparison with the baseline model, MKAD. This method of testing is typical in cases where multiple copies of data are expensive to train by the classifier models, hence the comparison on a single test dataset [21] as is the case in this work. With the McNemar's test, prediction results from two classifiers are transformed into a contingency table. The table is used to calculate the statistic and the p-value calculated by the test evaluates the significance of the classification.

A contingency table is a tabulation or count of two categorical variables. McNemar's test is interested in binary variables correct/incorrect or yes/no. This is a 2×2 contingency table. MKAD and Logic-OR are the two trained classifiers compared in this section. Each classifier makes binary class prediction for each of the examples in the dataset. The predictions are evaluated and determined to be correct or incorrect. The accuracy of both models on the dataset is retrieved and mapped to a contingency table. The contingency table relies on the fact that both classifiers were trained on the same training data and evaluated on the same data instances. The contingency table in Table 4-14 (a) below shows its general structure while (b) presents the contingency table count from the implementation in this work:

Table 4-14 McNemar Contingency Table structure with example sample counts

	Logic-OR Correct	Logic-OR Incorrect
MKAD Correct	Yes/Yes	Yes/No
MKAD Incorrect	No/Yes	No/No

(a)

	Logic-OR Correct	Logic-OR Incorrect
MKAD Correct	309	35
MKAD Incorrect	14	30

(b)

The 'Yes/No' label in Table 4-14 (a) reveals the count of those test instances that MKAD got correct but that Logic-OR got incorrect, and the 'No/Yes' count is the sum of test instances that MKAD got incorrect but Logic-OR got correct. In Table 4-14 (b), these values, as calculated from the evaluation in this work, are recorded. For completeness, the results for MKAD only are shown in Table 4-15 along with other metrics and results from different data combinations. The McNemar's test statistic is calculated as:

$$\text{Statistic} = (\text{Yes/No} - \text{No/Yes})^2 / (\text{Yes/No} + \text{No/Yes}) \quad (4-1)$$

It can be observed that only two elements of the contingency table are used, specifically, that the 'Yes/Yes' and 'No/No' elements, which show instances where they agreed, are not used in the calculation of the test statistic. The McNemar test is commenting on whether these two models disagree in the same way (or not). It is not informing on whether one model is more or less accurate or error prone than another.

The default assumption, or null hypothesis (H_0), of the test is that the two cases disagree to the same amount. If these cells have counts that are similar, it evidences that both models make errors in much the same proportion, just on different instances of the test set. In this case, the result of the test would not be significant and the null hypothesis would **not be rejected**). However, if these cells have counts that are not similar, it shows that both models, not only make different errors, but in fact have a different relative proportion of errors on the test set. In this case, the result of the test would be significant, and **the null hypothesis** (i.e. the alternative hypothesis) would be **rejected**.

The p-value is the probability of observing the McNemar statistic value meaning it is the probability that the outcomes from the data happened by chance. A p-value ranges from 0% to 100% but generally written in decimal. It is good to have low p-values. Lower p-values indicate the likelihood of the outcomes occurring by chance. For instance, a p-value of 0.01 indicates that there is only a 1% probability that the test outcomes occurred by chance. Generally, in many cases, a p-value of 5%, that is 0.05, is used as the significance threshold or alpha.

Given the selection of a significance threshold (i.e., alpha), the p-value calculated by the test can be interpreted as follows:

- **P-value > alpha**: fails to reject H_0 ; no difference in the disagreement; similar error proportions.

- **P-value $\leq$ alpha**: rejects H_0 ; significant difference in the disagreement; different error proportions

The McNemar's test in this work was implemented using Python. However, it was also carried out in MATLAB in order to ensure results are consistent. In Python, the *mcnemar()* *Statsmodels* function was used. The function takes the contingency table as an argument after generating the table using the *mcnemar_table* library and returns the calculated test statistic and p-value. The McNemar test was also implemented in MATLAB by calling the *testcholdout* function which "returns the test decision, by conducting the mid-*p*-value McNemar test, from testing the null hypothesis that the two predicted class labels have equal accuracy for predicting the true class labels" [22].

This section described the McNemar test statistical hypothesis to compare MKAD and the proposed Ensemble Logic-OR classifier models on a single evaluation dataset. The overall statistical performance comparison results between the baseline MKAD and the proposed Logic-OR ensemble is shown in the Table 4-15 along with the results from an integrated and separate combinations of the data types.

For the evaluation dataset with 388 flight examples consisting of both normal and anomalous flights, the p-value of 0.0026, as shown in the first cell of Table 4-15, reveals that the two models perform significantly different since it is less than the significant threshold of 0.05. Though MKAD and Logic-OR perform differently, the confusion matrix as earlier revealed, shows MKAD performing better on the overall evaluation data than the ensemble. This is because MKAD had less false alarms than the proposed ensemble meaning that the latter misclassified more normal flights as anomalous than MKAD did. However, when run only on a test dataset with the original anomalous flights only, the null hypothesis was still rejected (i.e., p-value = 2.9104×10^{-11}) but the ensemble is seen to perform better than MKAD with only 3 missed anomalies as opposed to 17 missed by MKAD.

Table 4-15 The combined results of statistical comparison between MKAD and Ensemble Logic-OR implemented on different data constituents.

Metrics	All Flight Data	Only Anomalous Flight Data	Only Normal Flight Data	Remarks
P-value	0.0026	2.9104×10^{-11}	1.5442×10^{-11}	<i>P-value shows significant difference between both models when run on all flight data as well as when the data is split into just normal or just anomalous flights. In all data use cases, the null hypothesis is rejected.</i>
TP MKAD	76	76		<i>MKAD is the baseline model. It finds less anomalies than the Ensemble</i>
TP Logic-OR	90	90		<i>Logic-OR is the Ensemble. It finds more anomalies than MKAD</i>
TN MKAD	268		268	<i>MKAD however correctly classifies <u>more</u> normal flights than the Ensemble</i>
TN Logic-OR	233		233	<i>The Ensemble correctly classifies <u>less</u> normal flights than MKAD</i>
FN MKAD	17	17		<i>MKAD misses <u>more</u> anomalies than the ensemble</i>
FN Logic-OR	3	3		<i>The Ensemble misses <u>less</u> (3) anomalies than the 17 anomalies MKAD missed.</i>
FP MKAD	27		27	<i>However, MKAD had fewer false alarms than Logic-OR</i>
FP Logic-OR	62		62	<i>Logic-OR had more false alarms than the baseline model</i>

Considering that the aim of this research is to build a framework that finds a wider range of anomalies which would, otherwise, be missed. The statistical analysis reveals that the anomalies found by the Logic-OR ensemble are statistically significant due to the low p-values. Also, the expert considered the ensemble results to be beneficial to the flight safety/analysis domain.

4.10 Conclusion

The results showed that the Logic-OR Ensemble AD method was able to detect more flights with operational significance than individual methods. This increased the chances of identifying more potential safety risks which are statistically significant as evidenced by the McNemar's statistical comparison test carried out in this work. The strengths and weaknesses of the Ensemble AD method were revealed from the operational characteristics of flights detected by its constituent detector algorithms. ClusterAD-Sample was generally more sensitive to instantaneous anomalies. The implementation of this algorithm by segregating data types, that is then handled by CAD-Cont and CAD-Cat, had a few benefits. One advantage is that the same parameter data types were handled together thereby bypassing the known issues with tackling heterogeneity by ML models. Having these two variants of the cluster-based algorithm also meant that the instantaneous anomalies detected were related to instant abnormal actions of the crew or spikes in the sensor information.

Image processing, like MKAD, highlighted anomalies present in both data types and was particularly beneficial in easily identifying some anomalous flights simply by visualising the processed image. While MKAD performed better in detecting anomalies in sequence of actions found within categorical parameters, it was, however, unable to detect anomalies associated with delayed crew actions even though they may have followed the expected sequence of actions. This ability was the unique property of CAD-Cat which gave the ensemble an edge. Anomalous flights detected by the Ensemble were more varied than those flagged by MKAD because the individual detectors complement each other in detecting different types of anomalies. Furthermore, the ensemble AD framework could also complement existing methods

by highlighting emerging issues that were previously unaccounted for. Flight manuals do not have information on every unique situation that the flight crew may face inflight and so being able to detect such emerging anomalous events could also help manufacturers and airlines improve the procedure guidelines for flight crew to increase their knowledge on the aircraft use.

References

1. (IATA), I.A.T.A., *Future of the Airline Industry 2035*. 2018, IATA.
2. Sheridan, K., et al. *An application of dbscan clustering for flight anomaly detection during the approach phase*. in *AIAA Scitech 2020 Forum*. 2020.
3. (FAA), F.A.A. *Safety Management System Manual*. 2015.
4. Administration, F.A., *Audit Report*. 2013.
5. (NASA), N.A.a.S.A., *Multiple Kernel Anomaly Detection (MKAD) sample dataset*. 2013.
6. Li, L., et al. *Anomaly detection in onboard-recorded flight data using cluster analysis*. in *2011 IEEE/AIAA 30th Digital Avionics Systems Conference*. 2011. IEEE.
7. Li, L., et al., *Anomaly detection via a Gaussian Mixture Model for flight operation and safety monitoring*. *Transportation Research Part C: Emerging Technologies*, 2016. **64**: p. 45-57.
8. Das, S., et al. *Comparison of algorithms for anomaly detection in flight recorder data of airline operations*. in *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. 2012.

9. Memarzadeh, M., B. Matthews, and I. Avrekh, *Unsupervised anomaly detection in flight data using convolutional variational auto-encoder*. Aerospace, 2020. 7(8): p. 115.
10. Igenewari, V.R., Z. Skaf, and I.K. Jennions. *A Survey of Flight Anomaly Detection Methods: Challenges and Opportunities*. in *Proceedings of the Annual Conference of the PHM Society*. 2019.
11. Budalakoti, S., A.N. Srivastava, and M.E. Otey, *Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety*. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2008. 39(1): p. 101-113.
12. Iverson, D.L. *Inductive system health monitoring*. in *International Conference on Artificial Intelligence*. 2004.
13. Bay, S.D. and M. Schwabacher. *Mining distance-based outliers in near linear time with randomization and a simple pruning rule*. in *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 2003.
14. Li, L., et al., *Analysis of flight data using clustering techniques for detecting abnormal operations*. Journal of Aerospace information systems, 2015. 12(9): p. 587-598.
15. Aggarwal, C.C., *Outlier ensembles: position paper*. ACM SIGKDD Explorations Newsletter, 2013. 14(2): p. 49-58.
16. Matthews, B., *Multiple Kernel Anomaly Detection (MKAD) Algorithm*. 2019, NASA Ames Research Centre: United States.
17. Miller, S., *FlightAnomaliesDetection*. 2017, Github: Moscow. p. ClusterAD-DataSample(aka Gaussian-Mixture clustering) anomalies detection method implementation.

18. Acharya, T. and A.K. Ray, *Image processing: principles and applications*. 2005: John Wiley & Sons.
19. Mohan, A. and S. Poobal, *Crack detection using image processing: A critical review and analysis*. Alexandria Engineering Journal, 2018. **57**(2): p. 787-798.
20. Michalski, R.S., J.G. Carbonell, and T.M. Mitchell, *Machine learning: An artificial intelligence approach*. 2013: Springer Science & Business Media.
21. Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural computation*, 10(7), 1895-1923.
22. <https://uk.mathworks.com/help/stats/testcholdout.html>

5 Case-Based Reasoning framework for isolating deviations from Standard Operating Procedure (SOP-CBR) in Flight Operations

This paper describes ‘SOP-CBR’, the architecture of a Case-Based Reasoning (CBR) framework for isolating flight cases identified as anomalous due to deviations from Standard Operating Procedures (SOP). Flight Anomaly Detection (AD) methods flag flight profiles that deviate from that of the majority as well as reveal the features where the anomalies exist. One of such anomaly types that flight analysts are interested in, is deviation from SOP by flight crew. This has been identified in the industry as a primary cause of incidents and crew training is constantly being improved to mitigate its risks. Training is however applied as a reactive solution to the problem. There is therefore a need for a proactive approach involving an automated system tailored to isolate anomalies that are due to procedure deviations. SOP-CBR is a framework that uses the power of reasoning to identify this type of anomaly. In this paper, the SOP-CBR workflow is described from receiving flight profiles from an ensemble anomaly detector (developed in previous work) to a case knowledgebase that categorises the profile as a specific anomaly type. This paper also demonstrates with flight profile cases, the capability of the reasoning system to become an expert system for isolating procedure deviations in flight operations. In addition to this, the study describes the ability of SOP-CBR to deal with the incorrectly classed edge cases from the ensemble detector. A future direction for expanding this framework is also discussed at the end of the paper.

5.1 Introduction

Across the aviation industry, the focus has been on using rule-based methods to identify abnormal events. This means that only known exceedances captured within these methods can be identified. This creates an opportunity for previously unknown abnormalities to be missed implying that they are not considered when putting in place mitigation strategies. With the aircraft system being developed to integrate more

systems, the complexity between them is also increasing. This means that combination of possible risks increases and in turn, becomes impossible to capture all combinations of anomalies in a rule-based approach. One type of risk that is identified during flight data monitoring is deviation from SOP.

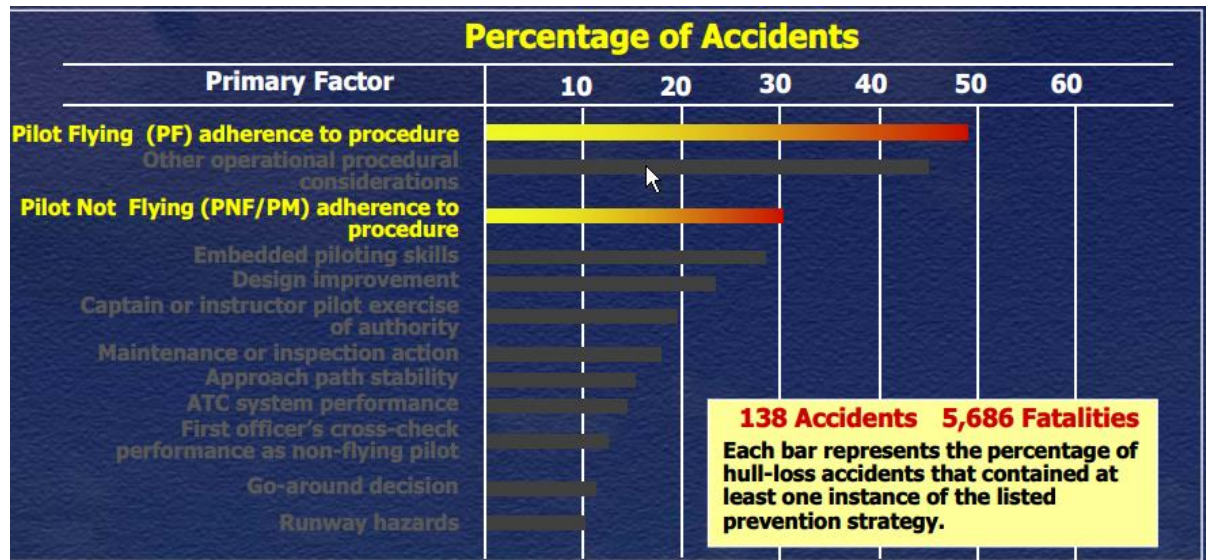


Figure 5-1 Percentage of hull-loss accidents with suggested prevention strategy [1]

Deviations from SOP by flight crew in the aviation industry is unfortunately, quite common and are currently revealed through flight analysis by experts. However, where such deviations are not detected, steps cannot be put in place to mitigate them in future flights and/or from escalating into an incident. In a 2013 presentation[1], a board member of the NTSB, Robert Sumwalt, revealed that addressing procedure deviations that occur within flights will address about 50% of accident causes in the industry (Figure 5-1). An example of an accident that occurred due to deviation from procedure is [2] where the pilot failed to engage the lift dump one second after touch down as the SOPs require. The 3.5 second delay resulted in the aircraft overrunning the runway and causing other damages to the aircraft. Other typical SOP deviation scenarios could also be: (i) non-use of standard phrases in messaging resulting in misunderstanding by the recipient (ii) a high speed or unstabilised approach which is continued to landing as opposed to the expected go-around thereby risking collision with the ground also known as CFIT (Controlled Flight Into Terrain). Some of these

procedure deviation events do not result in incidents in many flights hence, there is a need to identify these emerging subtle underlying confusions and errors of the flight crew to complement existing risk mitigation strategies and ultimately prevent the events from leading to much more severe outcomes [3].

5.2 Related Work

This work seeks to provide an automated system that can identify new hazards in form of procedure deviations without prior expert knowledge of the domain. In this section, the work describes, and reviews prior work carried out in terms of Flight SOP, traditional detection methods currently in use in the industry as well as the proposed reasoning approach to tackling the problem.

5.2.1 Standard Operating Procedures

Standard Operating Procedures for routine flight operations are documentation that describe how an aircraft should be flown in controlled airspaces as well as how it should be operated in the main phases of flight. The aim of having SOPs is to help ensure safety therefore, adherence to the procedures is expected with little variation but within tight tolerances. There are two types of SOPs [4]:

1. SOPs published by an aircraft manufacturer (e.g., Boeing, Airbus) designed to promote optimal use of the aircraft features. Airlines adopt these procedures and often customise it as their own organisational SOPs.
2. SOPs published within Air Traffic Control (ATC) units detailing how ATC officers are to carry out their responsibilities.

Flight crew are expected to adhere to the airline SOPs. However, studies [1, 5] have shown that non-adherence to SOP continues to be a primary cause of accidents today. Currently, flight data monitoring provides a way of generally identifying flight safety risks and serve as complementary approach to aid crew training in the highlighted areas.

5.2.2 Existing Methods

Conventional methods for monitoring and flagging risks such as procedure deviations in the aerospace domain are referred to as Exceedance Detection (ED) methods. Thresholds are set apriori within subsystems to reveal abnormal profiles. These exceedances are defined by three severity levels (level 1, 2 and 3) where level 1 highlights the exceedances with low severity and level 3 reveals those with the highest severity. The state-of-the-art techniques produce low accuracy and are limited to the domain for which they were developed.

A study [6] made use of data comprising of level 3 exceedance types like '*Speed Low at Touchdown*', '*Speed High in Approach (500ft)*' and '*Long Touchdown*'. It concluded that using tools that detect atypical behaviour complements the insights provided by ED methods. The work focused on revealing the impact of using discrete parameters and flight phases which capture crew actions thereby increasing the possibility of identifying procedure deviations. In another study [7], the authors adapt the process usually carried out by flight analysts on routine flight data by proposing a model of the flight crew activity. This meant that the work was able to highlight how the crew apply the SOPs and identify events that are closely related to that activity being analysed. This work also suggests the combination of data mining methods with its rule-based model to detect outliers originating from the modelled crew activity. Their work was extended in [8] where an XML (eXtensible Markup Language) was developed to model SOPs. XML is a coding language that defines a set of rules for encoding crew activity (in this case) in a human and machine-readable form.

However, detecting emerging risks with these approaches is limited to the crew activities that were modelled hence, it will need expertise to continuously update the modelled behaviour for an increasingly complex and dynamic environment such as flight operations. There is therefore a need for an automated way tailored to capturing deviations from SOP in routine flight operations observed within the flight data.

5.2.3 Case-Based Reasoning.

Case-based reasoning is a type of reasoning strategy that bases its learning on the use case examples it has observed over a period. The use case examples are stored in a case knowledgebase. The terms case database or case library will be used interchangeably in this work to describe the storage container of cases. This type of reasoning system generally comprises of sub blocks, namely: case recall, case adaptation, case evaluation as shown in Figure 5-2. These sub blocks are explained in the following paragraphs.

Case Recall:

This phase of the CBR system is aimed at discovering, retrieving, and selecting a relevant case to a new query case within the case knowledgebase. Many CBR systems establish their case recall step on attribute/feature similarity often carried out using weighting techniques in a nearest-neighbour approach [9]. The main steps involved in recalling a relevant case are Indexing, Retrieval and Selection. Indexing involves storing the cases appropriately in memory for effectively retrieving the similar cases [10]. The database search for cases is based on the index assigned to them. The index is based on a set of features of the case. The features for the new case becomes indexes into the database for which older cases with a subset of those features are retrieved [11]. In the selection task, the retrieved cases are evaluated based on the degree of similarity between these older cases in the knowledgebase and the new problem. The output of the case recall block is the case that is most similar to the query case profile as well as its stored solution from the database.

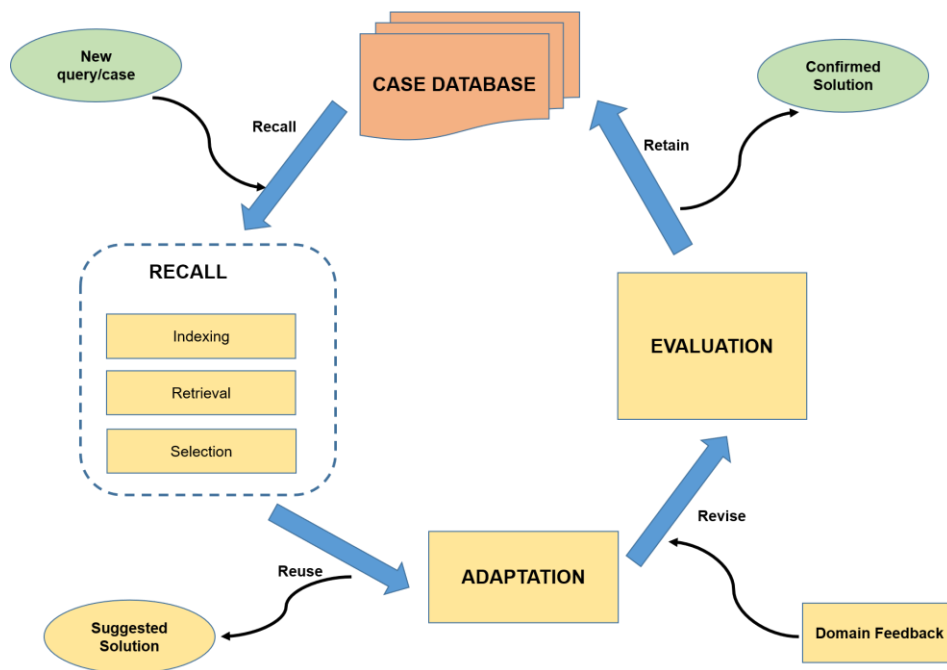


Figure 5-2 Case-Based Reasoning Process Flow (adapted from [12])

Case Adaptation:

After the selected case is recalled from the database, the focus is then set on adaptation. Adaptation in CBR is based on the premise that new cases hardly match existing cases exactly; there is therefore the need to fit an old solution to a new problem. In traditional reasoning systems, the solution assigned to the recalled case is adapted for the new case in the same way a human would have logically carried out. Adaptation also entails highlighting a new problem into the database in a situation where the reasoner does not retrieve any matches for the new problem.

Case Evaluation:

Evaluation of the suggested solutions, from the adaptation task, is carried out in this phase. The evaluation step is vital to any CBR system as it adds a level of expertise to the reasoning system. By applying solutions from the adaptation block to real-life problems, feedback can be provided at this stage. This feedback could be from an expert who would suggest solutions for the new problem after evaluating it. Without

feedback, the CBR system could repeatedly assign poor solutions to some new instances thereby reducing its effectiveness. The use case with its evaluated solution/outcome and any other useful information that the system can use [11] are then added to the case database as a new experience learned.

CBR has been implemented in many applications. It has been used in [13] as a strategy for prediction Explainability of deep learning models. The work combined a Neural Network model with a CBR system for explaining chronic renal disease. A similar study [14] was carried out on breast cancer data for explaining the classifications on new instances. The study made use of a CBR approach that has the capability of a classification algorithm as well as a visual representation of predictions for better explainability. A review of reasoning as well as recent implementations of CBR was carried out in [12] revealing the use of CBR in applications like text mining and intelligent systems for aircraft conflict resolution. A recommendation system was developed using CBR and implemented on bank client data for effective suggestion of personalised products and services to customers [15]. The authors conclude that the contextual information provided by the reasoning system assisted in improved user satisfaction.

Case-based reasoning is an established area of research. Considering that this application area is a safety-critical domain where missed anomalies could mean that the event is not addressed, continues to occur thereby possibly leading to an accident in the future. Therefore, it is essential that the proposed approach in this study can capture abnormalities that may have been missed by the Ensemble AD phase of this work. SOP-CBR framework is thus, proposed in this study to keep a case database of anomalies that have been sanctioned by an expert so that new instances of similar risk profiles can be described based on the anomaly type of previous use case examples. In other words, SOP-CBR is not only able to help identify anomalies that were missed in the AD phase but also categorises instances based on the anomaly type which in this case, would be SOP-deviated anomalies.

5.3 SOP deviation detection framework with Case-Based Reasoning (SOP-CBR)

The Ensemble AD framework precedes this reasoning phase and makes use of the Logic-OR technique to combine the results of individual AD methods. The outcome of the Ensemble AD is the identification of flights with anomalies and characterising the features where the anomalies exist. For a new instance, the ensemble decides if it is anomalous or not. There is, however, a likelihood that the ensemble does not class the instance appropriately whether by missing a true anomaly or by raising a false alarm for a flight that is normal. Since this uncertainty exists, the safety-critical application in this study requires an additional step where the new flight instance can be checked by an expert system to confirm as well as categorise its abnormality. This research is tailored to detecting the type of anomaly that reveals SOP-deviations and Figure 5-3 below describes how this is achieved.

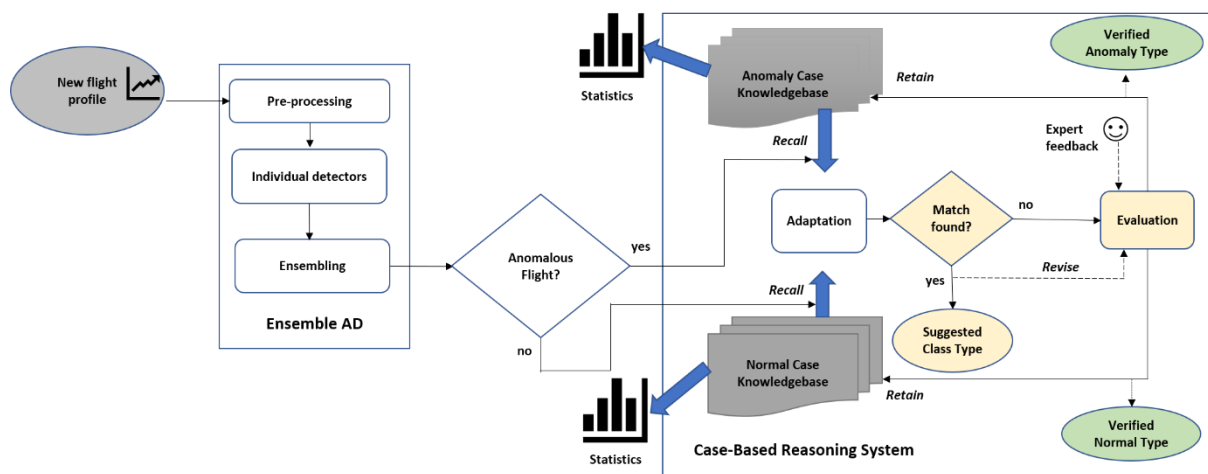


Figure 5-3 Overview of SOP-CBR Framework

The SOP-CBR workflow begins with a new/query instance being categorised as either a normal or abnormal flight by the ensemble anomaly detector. Based on how they are **indexed** within the system, the existing cases in the knowledgebase that match the attributes of the new instance, are found, **retrieved** and the most similar match is **selected**. This indicates the CBR system can **recall** relevant cases which are similar

to the query instance from the knowledgebase thereby also providing suggested solutions for the class type of the new instance. The K-NN (K-Nearest Neighbours) similarity technique is a supervised learning method used for classification and regression and assumes that similar things exist very close to one another. K-NN is used to discover and retrieve cases with similar profiles as the query instance. The anomaly type for the **selected** old case is then **adapted** to the new instance as its **suggested solution**.

The expert is presented with a Graphical User Interface (GUI) for **evaluating** the profile of the new case. This also allows the expert to provide **feedback** which is useful in effectively validating the suggested class type or categorising it where no match is found within the available use case examples. The verified class type is then **retained** in the **case knowledgebase**. For this safety-critical application, the workflow employs the expert to oversee the case library additions ensuring that every new case added, contributes significantly to the library. This way of learning helps increase the capability of the CBR by building effective case databases that cover the entire domain application and produce desired outcomes. Therefore, where a similar newer case is subsequently being analysed, the CBR system can match the attributes and recall the newly retained case so that the same anomaly type can be assigned to the newer instance.

Where the number of similar cases in the library for each class type reaches a certain population, the CBR stops retaining new instances of that type. Furthermore, there is a need to validate the expert system frequently (e.g. every 3 months) to ensure it is not repeating an error. This is carried out in the evaluation phase of the CBR workflow. The domain expert can carry out the evaluation periodically and is also flagged to provide feedback on new instances without a match from the case library. This framework was however, not implemented due to time constraints however, this work demonstrates its capability, flexibility, and the benefits it provides.

5.4 Demonstration Use cases

Usually, flight analysts would manually examine new query profiles to determine if they are abnormal or normal and if abnormal, they then need to ascertain the type of anomaly into which it can be categorised. Having to do this constantly and for many flights that occur daily in an airline, is time consuming. The Ensemble framework phase of this study is beneficial in that it would detect abnormal characteristics that differ from the majority while also identifying a wider range of anomalies than traditional detectors. This characteristic makes it more efficient than the conventional rule-based methods currently in use in aviation since it can identify emerging anomalies. The ensemble also outperforms its individual detectors in terms of the percentage of true anomalies it can identify. This research sought to primarily identify anomalies that are due to deviations from SOP hence, the introduction of the CBR approach.

In this section, this work seeks to demonstrate how this reasoning framework can be used to identify flight instances that are due to deviations from SOP. It is carried out by describing three scenarios and how they are used throughout the SOP-CBR workflow.

5.4.1 Scenario 1: A normal flight profile

Flight 91089 is a new raw flight profile which goes through pre-processing to reduce its features, ensure a 1Hz sampling rate and other dimension reduction steps as required by the individual detectors within the ensemble AD framework. The previous work describes this process in more detail. This flight profile is then served as input into each individual detector within the ensemble thereby producing individual labels for the flight. The combination step makes use of the Logic-OR technique to decide on the final classification of that flight as either normal or abnormal. It is also assumed that a high confidence in the output of the Ensemble, for normal class, can be inferred if all individual detectors agree on the class prediction.

In this case, the ensemble outcome for this flight case is considered normal as all detectors agreed on this classification. During this flight, the crew manually landed the flight and accurately set the aircraft into the appropriate configuration for landing at the appropriate time well before the final approach. Since it was confidently deemed a normal flight by the ensemble detector, the reasoning system simply retains this flight case in the **normal case library**. The SOP-CBR framework for this use case, gives a general categorisation to the flight profile so that other normal cases with similar profiles are annotated with the same type. The flight profile and its categorisation label information are retained in the normal case library.

In the rare cases where all detectors of an ensemble agree but incorrectly classify an anomalous flight instance i.e., missed anomalies or FNs, the evaluation exercise by an expert would help to reveal such false classifications. For a scenario where all the individual detectors do not agree on a label for a flight profile, the Logic-OR detector combination technique of the ensemble, classes that as an anomalous flight. This assumes that there must be an anomaly of interest to be investigated since the individual detectors have unique detecting strengths as described in chapter 4. The reasoning system deals with this possibility differently as explained in the next use case. This approach would cater for false alarms (False Positives) ensuring they are retained in the normal case knowledgebase.

5.4.2 Scenario 2: An Abnormal Flight with ambiguity

In this scenario, this study aims to describe the capability of the framework to categorise abnormal flights for which there is a high uncertainty about their anomaly type. An example of this is flight 58643. Its profile is the input into the individual detectors of the ensemble and its outcome classes this as an anomalous flight. This flight was only flagged by one of the ensemble detectors and the profile of the *NoseGear* (gear under the cockpit) feature was revealed as different from that of most flights in the data.

Aircraft gears cushion shocks and help with braking. The Nose gear specifically, is the gear located under the cockpit. It is used to steer the plane and is normally engaged during landing after the other two gears (left and right gears under the wings) have touched down. The ideal landing operation is that just prior to Touchdown (TD), the cockpit section of the aircraft is higher in the air than the tail due to the pitch attitude of the plane. This is the reason for the main left and right gears touch the ground before the *NoseGear* so that the tail wheels touch down before the cockpit wheels. The *NoseGear* should then be eased smoothly unto the runway. Holding the nosewheel off the ground for a longer period than normal, puts the flight at risk of slowing to an unusual speed that may in turn lead to the nose gear coming down too quickly. If this happens frequently, damage to that aircraft is inevitable.

Within the SOP-CBR workflow, the system checks this flight profile against older cases in the anomaly case library. Initially, the **anomaly case library** would start off empty and so, there would be no case with which to compare this query instance. The reasoning system then flags it presenting a GUI for the domain expert to evaluate and provide feedback. The feedback from the expert would be the type of abnormal operation e.g., late landing configuration or high-speed approach. During this flight, the crew embarked on a manual landing with full flaps and normal speed, thrust, pitch and wind levels throughout the window. However, it is discovered that the *NoseGear* was held off the ground for longer than expected as shown in Figure 5-5. The technique used by the crew in this case was to delay switching of the *NoseGear* from the 'Gear down-air' (represented as '2' in the plot) mode to 'Gear down-ground' (represented as '3' in the plot) mode. This meant that the *NoseGear* was held off the ground for about 6 seconds after the main gears were switched to 'Gear down-ground' mode. Since flights are allowed to be flown with little variation but within tight tolerances, then this profile can be considered as a deviation from the normal profile (Figure 5-4) though analysts may regard it as a low risk one. It is however an interesting deviation which would prompt an investigation to ascertain if it is a one-off scenario or if the specific pilot has implemented this technique frequently in the past. The investigation could explain why the flight crew chose the technique especially as other parameters seem to be in normal states. The expert could categorise this case as a '*deviation:NoseGearDelay*' reflecting the unique anomalous profile of the flight.

This flight instance and its category information are therefore, retained in the abnormal case library serving as experience gathered by the reasoning system for subsequent flight instances.

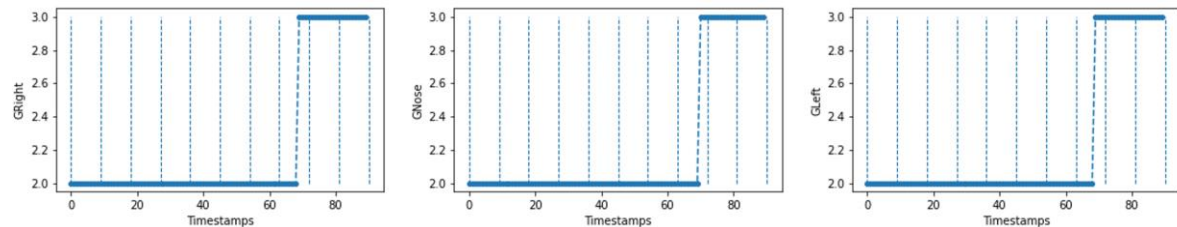


Figure 5-4 Normal Profile of Main and Nose Gear features covering final approach to post touchdown

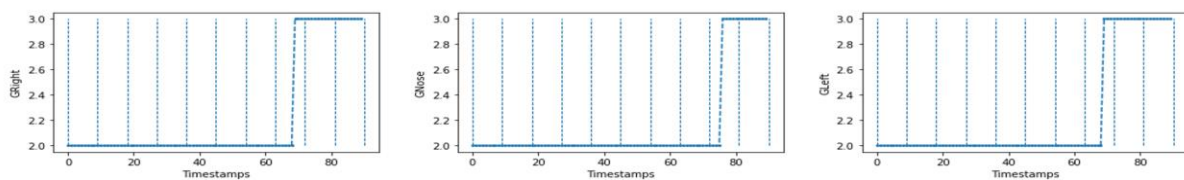


Figure 5-5 Abnormal Profile of Main and Nose Gear features covering final approach to post touchdown for 58643

5.4.3 Scenario 3: Abnormal Flight with older case examples

Flight 407681 was classed as an abnormal flight by the ensemble with anomalies in the pitch and thrust parameters. In this flight, the flight crew had landed poorly and so decided to make pitch and roll inputs during the rollout phase of flight. This was evidenced within the data specifically the pitch and roll sidesticks of the flight captain i.e., *CaptSSPitch* and *CaptRollSS*. The captain appeared to be over-controlling and using excessive inputs with full nose down input during rollout which is a poor technique to use. Furthermore, the SOP of the airline states that “*During rollout sidestick inputs (either lateral or longitudinal) should be avoided*”. Figure 5-6 presents a section of the landing procedure with this guidance.

Ground clearance

- Avoid flaring high.
- A tail strike occurs if the pitch attitude exceeds the angles shown below:

	A319	A320	A321	A330
Tailstrike	15.5°	13.5°	11°	17°
(landing gear compressed)	13.5°	11°	9.5°	11°

- A wing tip or engine scrape occurs if roll angle exceeds the angles shown below:

	A319	A320	A321	A330
Wingtip or engine scrape	18°	20°	18°	22°
(landing gear compressed)	16°	16°	16°	17°

- Be aware of a pitch up tendency with ground spoiler extension.

• **At touch down**

– **REV** **SELECT**

- Select REV thrust immediately after the main wheels touch down.
PNF will check on ECAM that REV thrust is selected on both engines.
PNF announces "NO REVERSE" if necessary.
- If runway performance is not limiting or where airport regulations restrict the use of reverse thrust, select and maintain REV IDLE until taxi speed is reached

Note: Where high brake temperatures may occur due to environmental conditions and brake fans are not available, REV MAX should be considered.

- If the runway is short or contaminated REV MAX should be used. Notwithstanding the above, REV MAX may be used at any time the crew deem it necessary in the interests of safety.
A slight pitch-up, easily controlled by the crew, may appear when the thrust reversers are deployed before the nose landing gear touches down.
Lower the nosewheel without undue delay.
- PNF continues to monitor the aircraft attitude.
- In case of engine failure, the use of the remaining reverser is recommended.
- Braking may begin before the nosewheel is down, if required for performance reasons, however when comfort is the priority, braking should be delayed until the nosewheel has touched down.
- During roll out, sidestick inputs (either lateral or longitudinal) should be avoided.
- If directional control problems are encountered, reduce thrust to reverse idle until directional control is satisfactory.
- After reverse thrust is initiated, a full stop landing must be made.

Figure 5-6 Extract of A320 aircraft procedures from the airline

This scenario assumes older case examples with similar profiles exist in the reasoner. Within the SOP-CBR framework, the flight profile is compared with profiles in the anomaly case knowledgebase. Each experience within that library possesses an index which enables the reasoning system to find it easily as described in section 5.2.3. K-NN then reveals flight cases that have a similar profile to that of flight 407681. A selection of the most similar case is made, and the category of that old case is extracted and presented as the suggested solution for that new instance. This suggested solution could be evaluated and validated as the verified anomaly type for this use case. Since this example is a deviation from SOP, the category could be 'SOPdev:ROutInput' to clearly differentiate this profile from other types of SOP deviations that the system may experience.

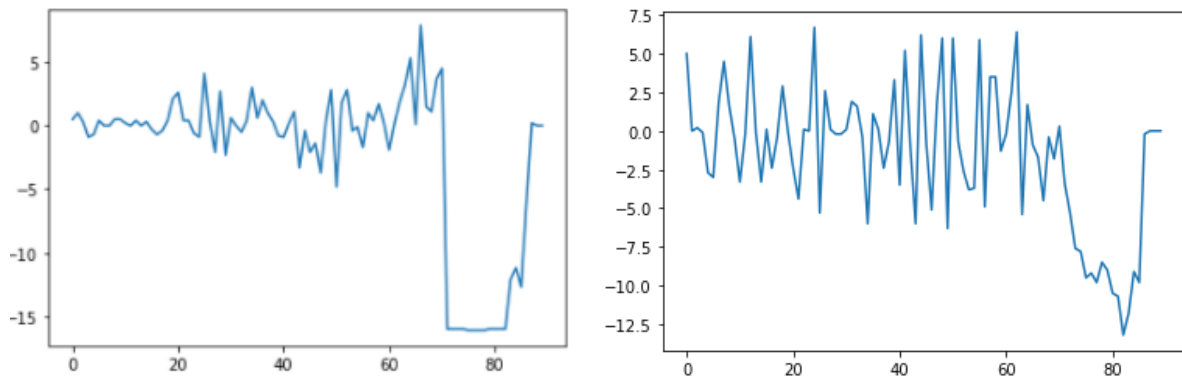


Figure 5-7 (left) *CaptSSPitch* (right) *CaptRollSS* for anomalous flight

5.5 Case-Based Reasoning for Statistics

The SOP-CBR starts out with an empty case database but as it learns from use case examples it receives, it grows over time, into an expert system. It eventually makes the work of flight analysts easier by providing accurate suggestions about the type of anomaly that constitutes abnormal flight instances. Specifically, it can isolate SOP-related anomalies, and this can be presented to the user as the solution.

As an expert system with many use case examples, the SOP-CBR is also capable of providing useful statistics on use cases it has learned over time. The framework can provide information about the number of case examples of a particular anomaly type it has encountered. It's ability to categorise use case examples with the same anomaly type makes it easier to determine if an anomaly type is typical behaviour for specific pilots. Statistics about the most common anomaly types can also be extracted. The aircraft family on which certain anomalies occur the most or least are also useful insights that can be gained from the proposed reasoning framework. Since the reasoner takes flight profiles that occur at any time of the year, the system can gain insights relating to seasonality i.e., to identify the time of year where specific event types recurred the most or the least.

A few of these statistics have been described below with examples.

5.5.1 The number of flight cases that constitute a specific anomaly type

In this dataset, there were only two (2) **‘go-around’** flights although there were several flights with unstabilised final approach configuration that should have resulted in a go-around. This highlights the fact that pilots for this airline chose to continue landing against SOP guidance thereby constituting a safety risk.

Several **dual side stick inputs** were also observed in the flight data. Side sticks are used for controlling the operation of an aircraft. On commercial jet planes like A320 used in this study, both the captain and First Officer (FO) each have side stick levers located close to them and are expected to follow basic task sharing principles during flight operations. In this work, the pitch and roll side sticks are examined, and it was observed, on some flights, that both the Captain and FO seemed to be controlling simultaneously. Ideally, in any aircraft operation, only one pilot should be flying at any given time and where the other pilot needs to take over the controls, there must be clear “I have controls” announcement. This indicates that there should only be input recorded from one sidestick at a time to prevent confusions. Dual input situations could sometimes contribute to operational incidents as was the case in the event described in [16].

The plot in Figure 5-8 shows the pitch sidestick inputs made between the Captain and the FO for the entire 90-second duration and for all flights in the dataset. The dots represent the flights at a given time and their location on the plot describes the position of the sidesticks for both pilots at that time. Where there are non-zero inputs from the captain (i.e., values on the x-axis), it is expected that there would be zero inputs from the FO. In other words, where the captain is in control of the flight throughout the window, the dots would fall on the horizontal line at input zero. Similarly, where the FO is controlling all through the flight window, the dots on the vertical line at zero would depict this situation. There are, however, situations where the task of controlling the aircraft in this period would need to be shared between both pilots. The same principle holds in that case too and the plots would still have the data points on either the x or y-axis at input zero.

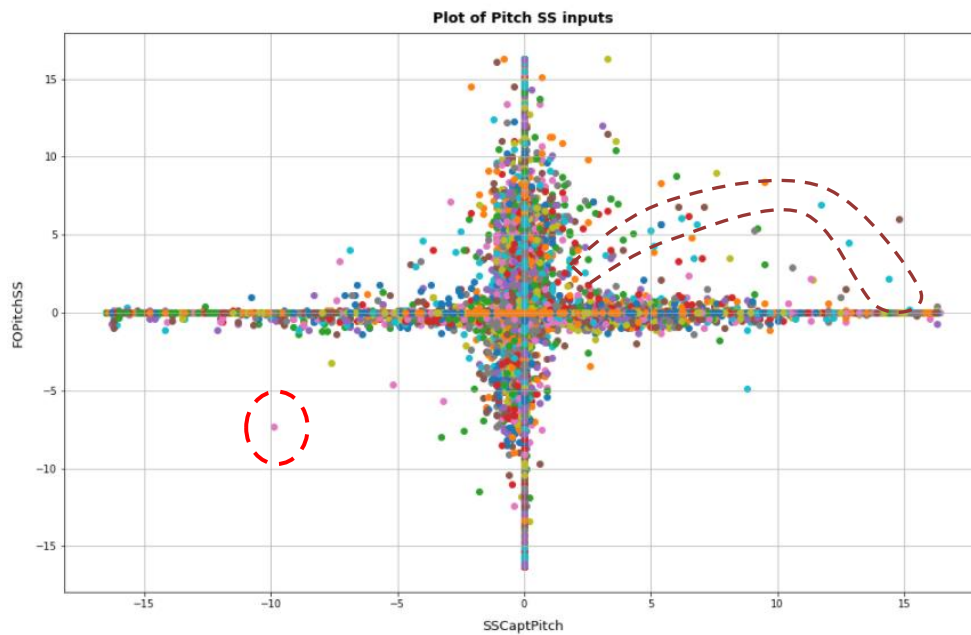


Figure 5-8 Comparing Pitch side stick inputs for captain and first officer

However, the fact that there are points away from the zero horizontal and vertical lines, means that there were double inputs made at some point on some flights during the landing window. For example, point (-10, -8) corresponding to (Captain input, FO input) is a point in time on that flight where both pilots were simultaneously controlling the flight. Though a single occurrence of this on a flight could be considered by analysts as a low-risk case or even noise. It was however, observed that on several flights (like that highlighted in blue on the plot) dual inputs occurred at multiple points in time. Dual input flight profiles had enough variation from the expected behaviour to flag them in the anomaly detection phase where the CBR is then able to appropriately categorise them. The pitch side stick plot in addition to the roll stick plot shown in Figure 5-9 show the population of flights with dual input anomalies in the flight data set.

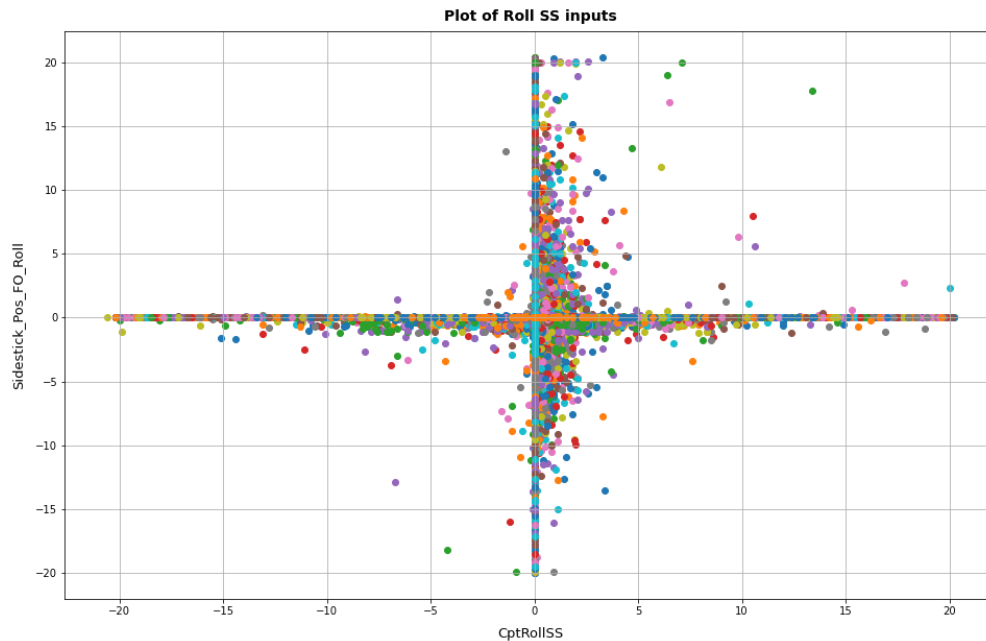


Figure 5-9 Comparing Roll side stick inputs for captain and first officer

Messy approach to landing: A handful of cases revealed some flight crew trying to capture glideslope from above in an unstabilised approach. Majority of flights entered the landing window in this study data at 800 ft above the ground. Figure 5-10 below shows one of the flights with this anomaly still at very high altitudes of 1200 ft and speeds at 180 kt compared to about 140 kt observed in majority of the flights on this route. This use case also reveals the flights as having fluctuating speed brake modes and ground spoilers deployed later into the window (Figure 5-11) indicating a struggle to control the flight during landing. The speed brakes on some of these anomalous flights were also left out until selection of FLAP FULL which automatically stowed it. The domain expert confirms that “crew would have been interviewed after this and would possibly be required to undertake training”.

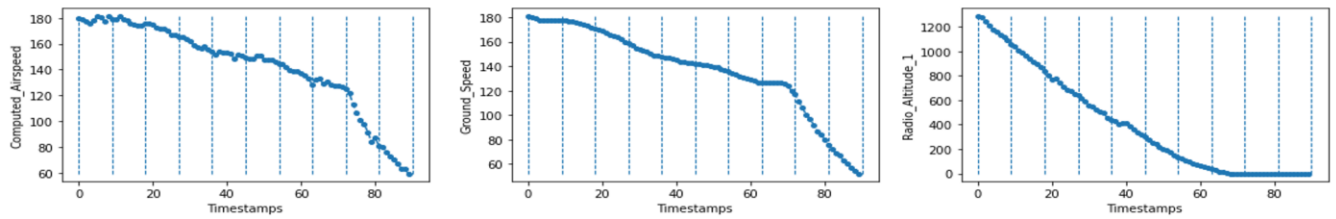


Figure 5-10 (left) Airspeed (middle) Ground speed (right) Radio Altitude profiles for anomaly case

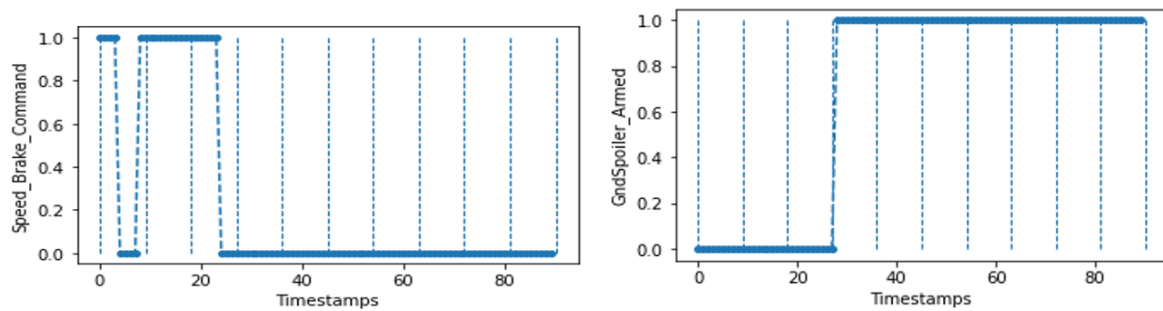


Figure 5-11(left) Speed brake mode (right) Ground spoiler mode

Thrust Reversers not deployed at touchdown: The SOP for landing indicate that thrust reversers need to be deployed immediately after touchdown (Figure 5-12) to help provide reverse thrust that bring aircraft to a stop. However, it was revealed that 102 flights in this dataset did not deploy the thrust reversers whether for Autoland or manual landing scenarios. This anomalous behaviour could cause the aircraft difficulty in stopping thereby leading to runway overruns which could in turn, cause injury and substantial damages.

- **At touch down**

- **REV SELECT**

- Select REV thrust immediately after the main wheels touch down.
PNF will check on ECAM that REV thrust is selected on both engines.
PNF announces “NO REVERSE” if necessary.
- If runway performance is not limiting or where airport regulations restrict the use of reverse thrust, select and maintain REV IDLE until taxi speed is reached

Note: Where high brake temperatures may occur due to environmental conditions and brake fans are not available, REV MAX should be considered.

- If the runway is short or contaminated REV MAX should be used. Notwithstanding the above, REV MAX may be used at any time the crew deem it necessary in the interests of safety.
A slight pitch-up, easily controlled by the crew, may appear when the thrust reversers are deployed before the nose landing gear touches down.
Lower the nosewheel without undue delay.
- PNF continues to monitor the aircraft attitude.
- In case of engine failure, the use of the remaining reverser is recommended.
- Braking may begin before the nosewheel is down, if required for performance reasons, however when comfort is the priority, braking should be delayed until the nosewheel; has touched down.
- During roll out, sidestick inputs (either lateral or longitudinal) should be avoided.
- If directional control problems are encountered, reduce thrust to reverse idle until directional control is satisfactory.
- After reverse thrust is initiated, a full stop landing must be made.

Figure 5-12 Extract of procedure for landing showing reverse thrust guidance

5.5.2 Anomaly types that were typical to a particular flight crew.

This was observed in a couple of cases with the nose down input in rollout shown in both pitch and roll sidestick features in Figure 5-13 for flight 175478 which was found to have been carried out by the same pilot as flight 407681 in Figure 5-7.

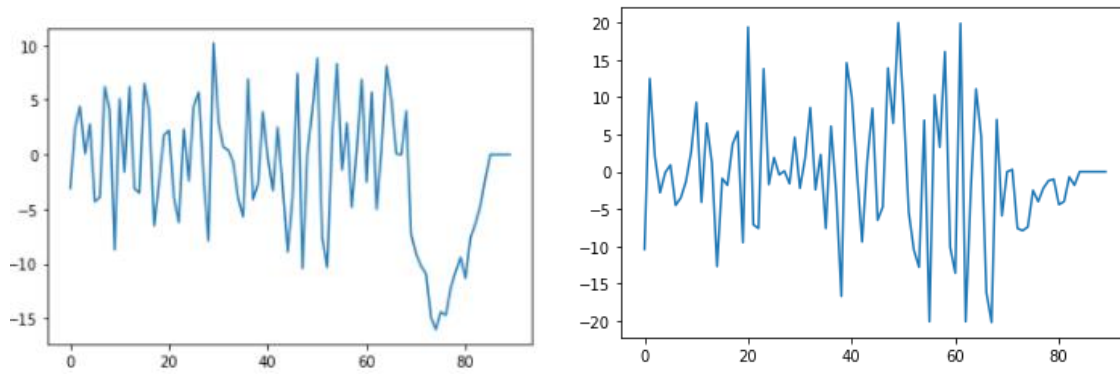


Figure 5-13 (left) CaptSSPitch and (right) CaptRollSS profile also showing similar nose down inputs after Touchdown in rollout

5.6 SOP-CBR Benefits and Limitations

The robust SOP-CBR framework proposed in this work provides several more advantages in addition to meeting the gap of isolating flights with SOP-related anomalies. The framework is flexible as it can be extended to identifying other anomaly types that may not necessarily be SOP related. For example, go-around flights are often flagged as an anomaly for further investigations. However, if the go-around procedure is performed properly, it will not necessarily fall under the SOP deviation category but regarded simply as a go-around. In other words, the framework is useful in explaining the outcomes of the anomaly detector. SOP-CBR is also scalable to accommodate multiple class types like the two class libraries described in this work.

The proposed framework complements crew training by providing context about the anomaly category that flight events belong. This work has focused on identifying flight risks highlighted due to procedure deviations. Crew training can therefore be more focused on specific flight crew and their typical training needs. The statistical insights SOP-CBR provides like the population of certain anomaly types in the case library, the most common anomalies, and poor techniques typical to certain pilots, helps airlines adopt a data-based training strategy.

The reasoning system used in this work is also beneficial for accurately identifying false classifications or edge cases from the ensemble detector phase. It also serves

as a means of dealing with flight misclassifications that may have arisen due to uncertainty in the median-based labelling framework described in previous work. In this work, periodic evaluations by the expert have been recommended to ensure the system is not repeating mistakes related to categorising flight cases.

Normalisation of deviance is a concept that refers to a process where a deviation from expected procedure has been adopted by a significant number of pilots so much that it is now perceived as being a normal behaviour. This usually occurs when such erroneous behaviour is missed, misinterpreted, or simply not corrected early on and can go on over time. In this research, there is a likelihood of these cases existing and is likely to affect the labelling and ensemble detection phases of this work. However, where a normalisation of deviance exists and use case examples with hazardous events have been stored in the normal case library, the SOP-CBR has the capability to detect it during the evaluation phase of the framework. A general training on those flight operation areas can then be prioritised to help improve the behaviour of the flight crew subsequently. The proposed framework complements the improvement of SOP guidance where these normalised events have been investigated and deemed to be safe [17].

The SOP-CBR framework can be computationally expensive where it must perform similarity checks on a large case knowledgebase especially in a two-case knowledgebase situation. The performance of the framework depends on the number of cases it stores in its library and the quality of the feedback it receives during evaluations. It is therefore important for the system to present the expert with an easy-to-use GUI to increase the quality of their feedback. Another area of future work would be to expand SOP-CBR by introducing a visual reasoning functionality into the framework in an efficient way. Visual reasoning explains the similarity results of the KNN technique by presenting a visual display of the shared characteristics between the retrieved older use case examples and the query case. Since, time constraints prevented the implementation of this framework, future work would also be in developing the visual interface and producing useful insights as discussed in this study.

Proposed SOP-CBR Visual Interface

The SOP-CBR framework depicted in Figure 5-3 highlights the need for expert feedback in categorising flight profiles of the new case; including validating the suggested class type or categorising it where no match is found within the available use case examples. It is proposed that a GUI or visual interface be presented to the expert to aid in achieving this. The GUI environment should comprise of a category recommendation element, a summary statistics aspect as well as other features that make the visual easy to use. The recommendation category aspect is an important part of the visual environment that guides the expert to logical categorisation of the new flight profile. The summary statistics aspect provides a frequency of SOP abnormal events which could help with flight analysis tasks and in prioritising or tailoring training. The additional menu features within the visual environment provides ease of navigation; these features could be the ability to load a stored case-base profile, update actions, tool tips to show an expanded information on visual charts, etc. A suggested category recommendation element for the visual environment is summarised in this section.

The CBR visualisation method implemented in [18] is also suggested as possible future work which can be adapted to the category recommender element of the SOP-CBR visual interface. This method integrates scatter plot graph with rainbow boxes [19] to produce a qualitative user interface for the expert to gain insights which would aid decision making. Scatter plots are widely used for visual classification by representing the distance matrix between the flight cases in two dimensions while rainbow boxes are a new technique that has only recently been applied to medical applications to display qualitative information that is lacking in common classification methods.

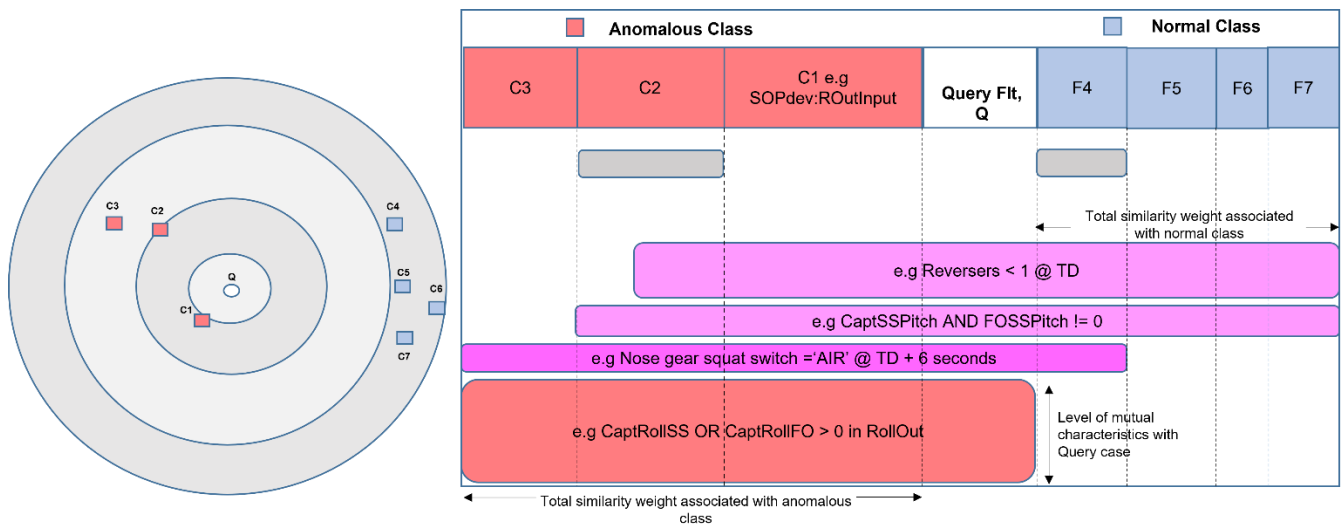


Figure 5-14 Proposed SOP-CBR category recommender interface adapted from [18]

Figure 5-14 is a suggested category recommender interface which would help a user decide which category a flight query belongs to. Since the colours depict the categories, the dominant colour determines the category of the query flight. On the scatter plot, the closer the points are, the similar the flight cases they represent. The rainbow boxes compare the query flight to the most similar cases belonging to the best candidate categories. The column width details how similar the specific case is to the query flight; the smaller the column, the less similar to the query and the taller the boxes the more mutual interests present between the case and the query. If a particular colour is more in number among the boxes and the width/height of the boxes are also greater, then the category for the query flight is deemed to be that represented by these dominant boxes. To include interactivity for the user, this integrated visual interface joins both methods together by presenting more information about a point on the scatter plot displayed by the appropriate rainbow boxes.

Figure 5-14 highlights the application of the suggested graphical visual interface to the flight dataset. The query flight case is input through the SOP-CBR framework with the scatter plot and rainbow boxes providing a visual to the expert for more informed decision making on the categorisation of the query flight, Q. The “normal” class was associated with blue in the figure, and the “anomalous” one with red. Seven (7) similar

cases to Q were extracted and boxes were also selected. In both visual plots, it can be observed that 4 similar cases were normal, while 3 were anomalous. The scatter plot reveals that C1, the most similar flight case was anomalous. Case C1 is shown on the rainbow box in column C1 with the largest width representing the most similar category label.

Rainbow boxes headers can be used to visually sum the weights of the similar cases and conclude that the weight of the anomalous cases is higher. Rainbow boxes show the common characteristics between Q and the similar cases. The most informative (bigger) box in the example is "*CaptRollSS OR CaptRollFO > 0 in RollOut*", and it includes all, and only, anomalous cases. Thus, the box is red. Other boxes are smaller, and purplish. Where some labels are not visible, the GUI can be designed for mouse-hovering which makes the labels more visible. In this example, the proposed visual interface argues strongly in favour of an anomalous flight classification specifically a 'SOPdev:ROutInput' category label. Moreover, the expert can mitigate their decision depending for example, on the importance given to '*CaptRollSS OR CaptRollFO > 0 in RollOut*' mutual characteristic.

In summary, the proposed graphical interface for the SOP-CBR framework visually aids the classification of the query flight case by using: (1) the scatter plot which shows the class associated with the closest similar cases, (2) the scatter plot and rainbow boxes headers then highlight the number of similar cases belonging to each class, and their similarity level, and then (3) the boxes indicate *how* the selected flight cases are similar to the query, and toward which class and category each characteristic value orientates. Therefore, this integrated approach allows both a *quantitative* and *qualitative* approach for visualising the classification.

5.7 Conclusion

Throughout this work, the architecture of the proposed SOP-CBR framework is presented, its life cycle described, and demonstrated using flight use cases. The benefits of the proposed framework are discussed, limitations are highlighted, an

integrated graphical visual interface is suggested as improvements proposed to gather information from the recall phase and present it during evaluation to aid the quality of feedback from the domain expert. With a fully experienced SOP-CBR, flight analysts would have an automated way of predicting if new flight instances are anomalous due to procedure deviations. Moreover, accident investigation organisations such as the NTSB and AAIB have processes that entail ascertaining if non-adherence to procedures was a contributing factor to accidents under investigation. This expert reasoning system would be useful in providing this information in an automated way as opposed to manual analysis often carried out today. The statistical insights retrieved from the system also ensures SOP-CBR can complement training strategies, procedure improvement as well as aircraft design enhancements which in turn increase flight safety.

References

1. Sumwalt, R.L. *Standard Operating Procedures: The backbone of professional flight operations*. 2013 [cited 2020 12.12.2020]
2. Wood, D.L. and T.T. Takahashi. *The Effect of Piloting Practices Upon Actual as Opposed to Scheduled Landing Field Performance*. in *2018 AIAA Aerospace Sciences Meeting*. 2018.
3. Yan, J., *Identifying opportunities of tracking major human factors risks through flight data monitoring*. 2014, University of Waterloo.
4. SKYBRARY. *Standard Operating Procedures (SOPs)*. SKYbrary Aviation Safety 2021 [cited 2021 13.12.2021]; Available from: <https://skybrary.aero/articles/standard-operating-procedures-sops>.
5. Jacobson, S. *Aircraft loss of control causal factors and mitigation challenges*. in *AIAA Guidance, navigation, and control conference*. 2010.

6. Maille, N. *On the use of data-mining algorithms to improve FOQA tools for airlines*. in *2013 IEEE Aerospace Conference*. 2013. IEEE.
7. Maille, N. *On the use of Flight Operating Procedures for the Analysis of FOQA data*. in *6TH EUROPEAN CONFERENCE FOR AERONAUTICS AND SPACE SCIENCES (EUCASS), Krakow, Poland*. 2015.
8. Maille, N. *Modeling airline crew activity to improve flight safety analysis*. in *2017 IEEE Aerospace Conference*. 2017. IEEE.
9. Biswas, S.K., D. Devi, and M. Chakraborty, *A hybrid case based reasoning model for classification in internet of things (iot) environment*. *Journal of Organizational and End User Computing (JOEUC)*, 2018. **30**(4): p. 104-122.
10. Rong, Y.K. and S. Huang, *Advanced computer-aided fixture design*. 2005: Elsevier.
11. Kolodner, J.L., *An introduction to case-based reasoning*. *Artificial intelligence review*, 1992. **6**(1): p. 3-34.
12. Ezhilarasu, C.M., Z. Skaf, and I.K. Jennions, *The application of reasoning to aerospace Integrated Vehicle Health Management (IVHM): Challenges and opportunities*. *Progress in aerospace sciences*, 2019. **105**: p. 60-73.
13. Vasquez-Morales, G.R., et al., *Explainable prediction of chronic renal disease in the colombian population using neural networks and case-based reasoning*. *IEEE Access*, 2019. **7**: p. 152900-152910.
14. Lamy, J.-B., et al., *Explainable artificial intelligence for breast cancer: A visual case-based reasoning approach*. *Artificial intelligence in medicine*, 2019. **94**: p. 42-53.
15. Hernandez-Nieves, E., et al., *CEBRA: A Case-Based Reasoning Application to recommend banking products*. *Engineering Applications of Artificial Intelligence*, 2021. **104**: p. 104327.

16. COMBES, F. *Dual Side Stick Inputs*. Airbus Safety First, 2006.
17. Schmidt, T.A., H.K. Kourdali, and J. Nixon, *Evaluating process-based and crew-centred approaches to procedure design in aviation: Workload and performance changes in go-around manoeuvres*. *Applied Ergonomics*, 2021. **90**: p. 103244.
18. Lamy, J. B., Sekar, B., Guezennec, G., Bouaud, J., & Séroussi, B. (2019). Explainable artificial intelligence for breast cancer: A visual case-based reasoning approach. *Artificial intelligence in medicine*, 94, 42-53.
19. Lamy, J. B., Berthelot, H., Capron, C., & Favre, M. (2017). Rainbow boxes: a new technique for overlapping set visualization and two applications in the biomedical domain. *Journal of Visual Languages & Computing*, 43, 71-82.

6 Discussion & Summary

The aviation industry has focused on safety enhancement strategies for decades to maintain their stellar track record for safety. The predicted increase in air travel means that if safety levels remain constant, there will be a corresponding increase in accident rates. Since accident rates are used as a measure of safety levels in the industry, there is a need to examine its major causes with a view to proffering solutions that could, in turn, reduce the rates at which accidents happen. Human errors make up 80% of accident causes. These errors could be by engineers, Air Traffic Controllers, and flight crew.

Furthermore, the primary cause of flight incidents is non-adherence to SOP. These procedures are in form of guidance that help the flight crew maintain safe operations of the aircraft they fly. An increase in SOP adherence by flight crew is said to be the main strategy for reducing accident rates thus increasing air safety [1]. Training is a common way flight crew are provided with useful knowledge that helps them make good decisions thereby ensuring they are operating the aircrafts safely. Training is, however, reactive as it is often planned around accidents or incidents that have already occurred. Today, the stakeholders in the industry agree that reactive systems should be complemented by proactive systems that use data for improvements. Therefore, aviation has shifted to a proactive method of using data to detect events in advance so that mitigation plans can be put in place to prevent similar risks in future.

Currently, the industry uses rule-based methods to identify risk precursors (such as procedure deviations) in flight operations as they flag events that exceed predefined thresholds. The use of thresholds that have been set in advance indicates that this method will miss events that have not been predefined. There is a high possibility of the exceedance technique missing important events considering the frequent introduction of sub systems into the aircraft design thereby leading to more complex interactions that in turn foster more risky combinations.

Recently, the Flight Safety Foundation published a paper [2] encouraging the aviation industry to wholly accept a data-driven approach to pilot training. In the release, they recommended data-driven training programs that are developed and updated based on competency or evidence and not only based on flight hours of the pilots. In other words, evidence of pilot actions mined from within flight data should drive their training. In a safety-critical environment such as aviation, systems that detect emerging anomalies within flight data is sought especially those that can automatically identify anomalies due to procedure deviations.

Literature reveals several Anomaly Detection methods [3-8] that can highlight emerging procedure deviations that have not been predefined. These AD methods work on the assumption that the majority profile within the data represents the norm and so flight profiles that fall outside that norm are flagged as abnormal. Previous work describes the strengths and limitations of these individual AD techniques. These limitations indicate that there are types of anomalies that they are unable to detect. There is however also a lack of focus in specifically detecting procedure deviations which highlight actions and inactions of the flight crew that differ from that of the laid down operating instructions. A Framework for Anomaly Detection of flight-crew deviation from Standard Operating Procedure (FAD—SOP) has been proposed to identify anomalies within flight data that are due to procedure deviations by using a data analytics approach. This work is important as these procedure deviations often lead to unstable situations in-flight which undetected and unaddressed could eventually lead to serious incidents in subsequent flights.

FAD-SOP is a robust framework which employs the use of data from Flight Data Recorders in a novel integration of an ensemble AD strategy and a case-based reasoning system demonstrating flexibility and adaptability to and taking advantage of new requirements. The framework applies three main techniques: Multiple Kernel Anomaly Detection (MKAD), ClusterAD (Clustering Anomaly Detection) and Image Processing in its AD ensemble and then a reasoning system that isolates those events that are due to procedure deviation. A few examples of events identified are described as evidence of FAD-SOP's capability to identify more abnormal events than the

traditional AD methods, highlight causes, and specifically identify which of the anomalies reveal procedure deviations.

This chapter summarises this thesis work and highlights the important contribution to knowledge carried out within the three-year research period. This research explains how FAD-SOP integrates the power of reasoning with ensemble AD to enable identification of procedure deviations from aircraft data. The subsequent sections describe the FAD-SOP framework and its evaluation. Therefore, section 6.1 summarised the contribution to knowledge as evidence for completing the earlier-stated objectives. Section 6.2 informs the general assumptions made throughout this research thereby providing the scope and perspective followed by the author. In section 6.3, an evaluation of FAD-SOP is carried out by outlining its benefits, limitations, and future work in advancing its design. The section also describes the implications of this research, challenges faced as well as examples of deviations observed throughout the study. The summary and conclusion of work carried out is provided in section 6.4.

6.1 Contributions to Knowledge addressing Research Aim & Objectives

This chapter brings together work carried out in a three-year project to build FAD-SOP which is the overall aim of this research. This aim is achieved by fulfilling a few objectives described in chapter 1. The section describes the labelling framework and how it was complementary to training and evaluation of core parts of the framework. The pre-processing steps carried out on the data are then introduced, the Ensemble AD described and the Reasoning blocks for FAD-SOP briefly explained. These constituent building blocks of FAD-SOP serve as the contribution to knowledge as well as evidence the fulfilment of the research objectives.

The following sections describe the building blocks that make up the framework. **Error! Reference source not found.** shows an overview of these FAD-SOP building blocks in a workflow format:

1. The Data Labelling
2. The Ensemble Anomaly Detection
3. The Case-Based Reasoning (CBR) system for SOP deviation detection

This research sought to address the limitations of current flight AD methods by combining their strengths into an ensemble to reduce the number of undetected events which were the limitation of the individual methods. On identifying a wider range of anomalies, they are processed by a reasoning system to identify the anomalies that are due to SOP non-adherence. Explainable AI suggests that black box models (like the ensemble) are complemented by white-box systems (such as CBR) for better understanding and interpretation of the prediction outcomes [9]. The Ensemble phase is important because it reduces the computational work required by the reasoner since it is developed to classify the flights as either normal or abnormal. Among the abnormal flights, SOP-deviation events can be isolated. Usually, the categorisation as a particular anomaly type would have been carried out manually by the domain expert. However, over time, the reasoner becomes an expert as well and carries this out automatically making it less time consuming and laborious for the flight experts.

6.1.1 Data Pre-processing

Data pre-processing is the start of the framework implementation workflow in **Error! Reference source not found..** Chapter 2 fulfilled the first Objective as summarised in Table 1-1 since it described the evolution and functionality behind FDR data and how they are collected for analysis. It was shown that the data in its raw form is heterogenous and needs to be prepared for further use by ML algorithms. The raw flight data forms the input into FAD-SOP and in addition to its homogeneous nature, it also contains missing data, irregular sampling rates across parameter quantities and a significant number of parameters. Data for a single flight is represented in the form of a matrix and includes time samples as the row entries as well as the columns representing the recorded FDR parameters.

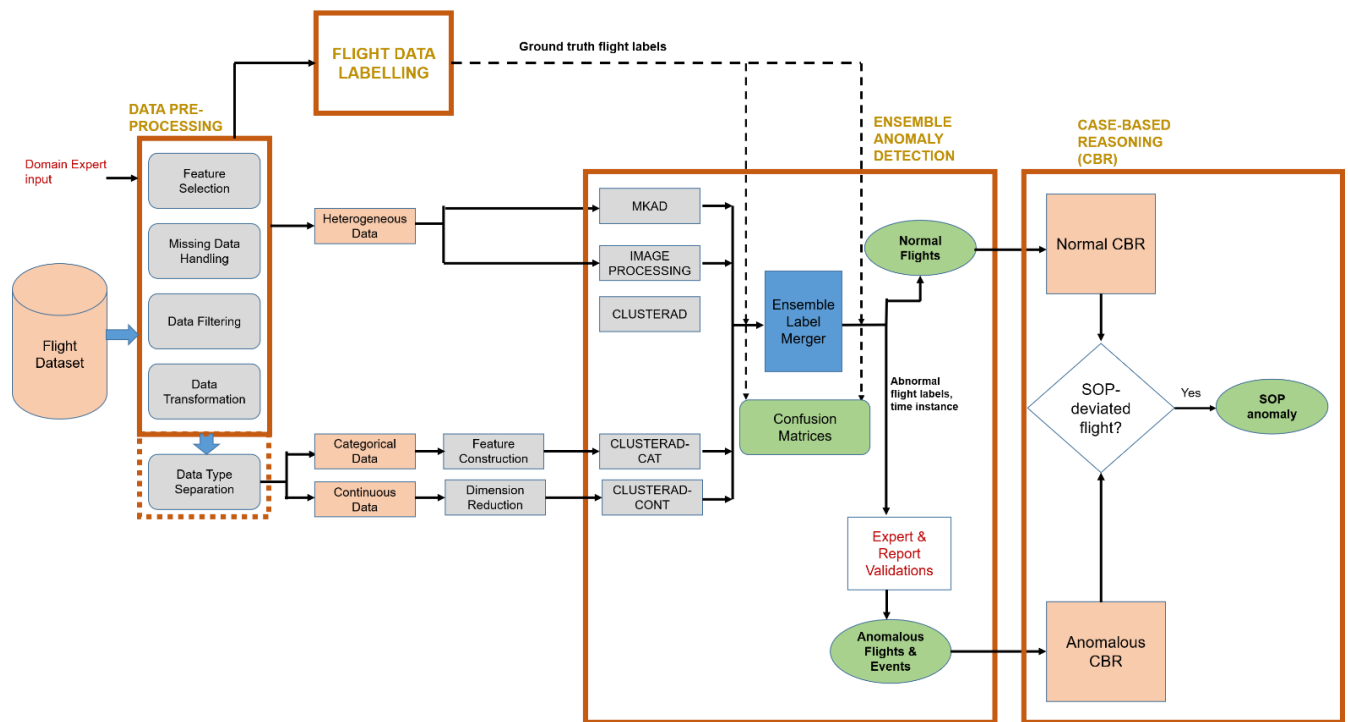


Figure 6-1 FAD-SOP General Overview

The data preparation module carries out operations such as feature selection, data type segregation, dealing with missing values and data transformation. All flights used in this study were from the same aircraft fleet, same route flown (though not runway-specific), and all flights focused on the same landing phase time window. This helps ensure the flights are comparable and the context behind the final outcomes can be better explained and understood. The **feature selection** entails choosing the parameters to work with throughout the process and is often aided by domain experts depending on the type of study to which it is being applied. Since different detection techniques require the data in specific form before it can be usable for them, the preparation step also included **separating** the continuous and categorical data types so they can be applied by the algorithms. There was a need to **filter** for noisy and **handle missing data** caused either by sensor or recording errors. With the help of domain expert, the **data transformation** block created new parameters from a set of parameters providing the ability to track more specific events within the data. For example, the recorded pressure altitude values were transformed to values relative to the airport such as height above touchdown. Dimension reduction of the data was also necessary for best utilisation of data to get the best model but was carried out at a

different point within the framework. The data pre-processing building block generally produces a time series representation of the continuous parameters along with the time series for the categorical variables and then provides the combined heterogeneous form. Specific details about this block are provided in Chapter 3.

6.1.2 Labelling Framework

Labelling phase was core to this work because it produced the labels for the partially labelled data set which was available for the study. This was also covered in chapter 3 and met the second objective. A lot of time was spent here because only 13 flights were initially labelled, and this work had to proffer a labelling solution that could make best use of these pre-labelled flights to label the entire dataset with over ten thousand flights. Besides, the normal flights used to train the detectors in the Ensemble AD phase were extracted from the outcomes of the labelling framework. The outcome of individual AD methods as well as that of the ensemble method are evaluated by comparing their labelling output with known ground truth labels of all the flight instances. This would form the basis for statistically ascertaining how well these methods were able to identify anomalous events in flight data and in turn how accurate their flight annotations were. A confusion matrix was used to show this information. Since the annotation information was initially not available in this application, there was a need to begin the methodology implementation with an unsupervised approach to labelling which reduces the manual and expensive labelling task often required from the domain experts. The median-based labelling technique proposed, assigned labels to all the flight instances used thereby providing the information required to evaluate how individual or ensemble AD methods performed. Therefore, this was a critical and significant step in the implementation phase as errors in the labels here, could mean an unfair comparison is made regarding the performance of the AD methods used. For domain stakeholders to have confidence in the labelling outcome and make informed decisions based on its output labels, it was necessary to include an estimation of the uncertainty that likely exists in the labelling of the flights. Chapter 3 explains this labelling framework in more detail.

6.1.3 Ensemble Anomaly Detection

Once the data has been prepared and labelled, the flight AD and other algorithms can be applied. This work considered three main algorithms for the detection: Multiple Kernel Anomaly Detection (MKAD), ClusterAD and Image Processing. Due to the conclusion in [10] that ClusterAD performs better on continuous data types than on categorical parameters, this study split the ClusterAD algorithm into two namely, CAD-Cont (ClusterAD applied on just continuous time series data) and CAD-Cat (ClusterAD applied on categorical data only) to improve on the performance. This resulted in a total of four individual algorithms that formed the core of the ensemble AD framework with MKAD being the baseline. MKAD was retained as a part of the Ensemble because it gives a degree of coverage albeit not a total coverage of anomalies. Chapter 4 evidences the fulfilment of the third objective providing more details about why these algorithms were selected, how they function, and their statistical and contextual performance evidenced in their output. The output of all four individual detectors is similar in that they inform which flights have been flagged as anomalous although, they differ in how they inform about the causal factors within the anomalous flights. Hence, it was easier to combine these algorithms via their individual label outputs.

The input into the ensemble merger were the labels from the independent techniques. Several ensemble strategies were evaluated by Logic-OR, which identified the most abnormalities was selected as the final ensemble technique for this work. The final list of anomalies from the chosen ensemble method identified more varied anomalies than the individual techniques and outperformed the baseline algorithm, MKAD. It is then assumed that finding a wider array of anomalies at this stage ensures that more significant procedure deviation anomalous events are identified in the next phase of the framework than would have been possible with an individual detection method. Statistically, a confusion matrix is computed for the ensemble using the ground truth labels from the data labelling phase. The matrix shows the proportion of flights that were correctly labelled as anomalous or normal and then the proportion that were misclassified as either normal or abnormal for the ensemble. This work assumes a bias towards detecting true anomalies/positives than true normal/negatives because flight analysts prefer to deal with a relatively large number of false alarms than to miss

a potentially hazardous anomalous event. These expectations and assumptions formed the premise for the ensemble choice which then serves as input into the SOP-CBR block of the FAD-SOP.

6.1.4 SOP-CBR

Where new flight instances are served as input into the ensemble, the class output and flight profile are in turn transferred into the reasoning stage shown in **Error! Reference source not found.** The work involved in this reasoning phase of the research fulfilled the fourth objective. The type of reasoning system proposed in this work is referred to as Case-Based Reasoning whereby knowledge from similar use case examples is adapted to provide a solution for a new problem instance. First, the **recall** phase, selects one or several similar cases from the case base (database of all flight cases). In the subsequent **reuse** phase, the solutions contained in those cases are **adapted** according to the profile of the new query. In the **revise** phase, the solution determined so far is verified in the real world and possibly corrected or improved, often by a domain expert. Finally, the **retain** phase takes the feedback from the revise phase and updates the knowledge of the reasoner, particularly the case base [11]. Inspired by this architecture, this work proposed a CBR system framework design that can classify a new flight fed from the ensemble as either SOP deviation or not based on the classification of similar historic cases. With SOP-CBR, similarity matches help determine if a new flight matches flight instances/cases already existing within the case database. Where a match is found with similar case profiles within the knowledgebase, the new instance can also be assigned the same anomaly type thereby showing the validity of FAD-SOP to isolate specific anomaly types. The initial cases are built with the help of an expert however, subsequently, an expert is only required for periodic evaluation and retaining of new use case examples within the case database. The proposed design framework includes a visual interface showing the parametric profiles of the new flight and those of its similar case instances from the case database. The visual aids are expected to give useful information to the expert about these cases so that informed categorisation decisions can be made where the reasoner does not find a relevant match to work with.

6.1.5 FAD-SOP Demonstrations

In chapter 4 and 5, several demonstrations, using real use cases from the data, were provided verifying the robustness, adaptability, and scalability of FAD-SOP. The fifth research objective was met through these demonstrations as summarised in the subsequent paragraphs.

First, FAD-SOP's robustness is shown in its ability to identify more anomalous events in the framework than the baseline method. In addition, the demonstration showed the characteristic of the framework to detect emerging anomalies which do not conform to the normal profile. It also showcased the ability to highlight more outlier events than the individual AD methods.

Its adaptability is demonstrated in its identification of new SOP anomaly profiles in addition to its ability to detect those procedure anomaly types previously observed and stored in its reasoning database. The framework is not only able to identify the majority normal profile but could also be adapted to categorise different types of normal profiles.

The framework uses a knowledge base to isolate root causes that are specifically due to procedure deviations. FAD-SOP's scalability can be extended towards isolating other causal factors in addition to procedure deviations.

6.2 Research Assumptions

In this work, several assumptions were made. Firstly, it is assumed that every anomaly type is important to highlight for investigation. The reasoning behind this assumption is that anomalies can indicate potential risks that are critical to flight safety. Also, anomalies could highlight aircraft or operational control failures and flight safety improvement opportunities.

Secondly, it is assumed that the majority of flights in the data available in this research represent a normal safety profile. However, it should be recognised that there are

varying types of normal. It is also assumed that a sample of the available flight data has been pre-labelled by the domain expert according to different conditions of normal and abnormal flights. Finally, it is assumed that flight profiles with no associated labelling can be categorised by a combination of supervised and unsupervised machine learning techniques. These assumptions were useful in streamlining the focus of FAD-SOP and its expected results.

6.3 Evaluation of FAD-SOP

6.3.1 Benefits of proposed framework

Though this work focused on a 90-second landing window around the touchdown point, FAD-SOP could also be adapted to any flight phase that can be described using a reference point in time and have a substantial amount of human-machine interaction. For example, take-off phase and approach. This ensures that all flights have the same length and so are comparable. It also shows the flexibility of the framework. The framework can be scaled to include other combination of detectors in the ensemble other than those used in this work. Also, with the ensemble AD phase of FAD-SOP, there is a benefit for flight analyst as they are presented with more anomalous events that happen in flights than they would have got from traditional methods. The ease of updating the knowledge of FAD-SOP is one major strength of the framework as new use case examples regarding anomaly types can be retained in the case knowledgebase. FAD-SOP is a hybrid of expert systems with data-driven methods thereby increasing the dependency of its case base and derive more information from existing data as opposed to full reliance on experts. Since the reasoning behind FAD-SOP is developed on an integration of frameworks, misclassification rates from the Ensemble AD phase are inherited by the SOP-CBR. However, the reasoner's periodic evaluation can highlight such misclassifications and the case retained in the appropriate case library as a learned use case example to be used subsequently. Statistical insights can be gathered from the use case examples of FAD-SOP thereby providing answers to questions like: 'how many flights deviated from SOP?', 'What anomaly type was the most or least common from the experiences learned by the

reasoning system?', 'where their abnormal behaviours that were typical to specific pilots?'

6.3.2 Disadvantage of proposed Framework

Positioning features were recorded in a resolution that made some anomalies unclear. An example was the deep landing which could only be observed by physically examining the gradient on the Radio Altitude parameter as described in chapter 3. Requirements for parameters with high resolutions and recording frequencies should be considered within the data [12] to further improve the performance of FAD-SOP. It is acknowledged that this work is limited to insights provided by FDR data from aircraft sensors alone. Therefore, other external factors that could have contributed to the anomaly are not considered in this work. Such factors are weather, traffic control instructions, bird strikes, etc.

6.3.3 Challenges/Limitations

Labelling the large number of flights used in this work was the first major hurdle that was surpassed. A considerable amount of time was spent in developing a labelling framework that is fit for a safety-critical domain. Since the flight labels were important for deciding on training data to use as well as target label information for evaluating detectors in the ensemble phase, it was important that the process produce highly accurate outcomes.

Normalisation of deviance was a possibility that was considered in this research. This concept refers to a process where a deviation from expected procedure has been adopted by a significant number of pilots so much that it is now perceived as being a normal behaviour. This usually occurs when such erroneous behaviour is missed, misinterpreted, or simply not corrected early on and can go on over time. In this research, this was also considered during the labelling phase as it would be counterproductive to label flights that fall in this category as normal simply based on their high population within the data.

Edge cases were also deemed a challenge encountered in this work. These are events that fall in the boundary region between normal and abnormal class. If analysed in one way, could be annotated as normal and vice versa. Edge cases occur rarely and in most machine learning methods are treated as noise and ignored. However, for a safety-critical application, this research chose to highlight these rare occurrences as anomalies requiring further investigation.

The FAD-SOP framework requires an expert to sentence anomalies that have been flagged thereby validating the results. It should, however, be noted that although a domain expert was used, it could also be seen as a limitation in this work as it reduced validation speed due to the expert not always being available when needed. This research tried to workaround this by seeking lots of general clarification from other pilots online albeit they had no access to the dataset. This helped move things forward until the main expert could confirm or disapprove the conclusions. Within the process, flight profiles were also condensed to a representative sub-set for validation so that the expert only had to work with a smaller sample set which would still provide a comprehensive level of label coverage. In addition to this limitation, the framework made use of only a single expert for validation of results. This posed the challenge of a single point of failure. Therefore, it is recommended for future studies, that a wider body of experts be used to validate the workflow in real life.

The framework proposed in this research included the use of just one aircraft type, one route and one operator. Hence, further work should consider other aircraft types over varying routes. Also, both domestic and cargo type operators would be useful to incorporate in future workflows.

6.3.4 FAD-SOP Implications

One of the implications of this research is that it is able to complement general or specific training of the flight crew by using the results of and the insights gained from FAD-SOP to improve training outcomes. Since SOPs are unable to cover every unique situation that pilots face during routine aircraft operations, being able to detect procedure deviations help manufacturers and airlines improve the procedure

guidelines for flight crew to increase their knowledge on the aircraft use, where required.

The higher number of false alarms could be regarded as a cost to flight analysts. However, dealing with these is already a part of the FDM process and so, flight analysts do not regard this as much of an additional cost if there is evident benefit in detecting significant hazards than current methods provide. Also, accident investigation organisations such as NTSB and AAIB can make use of FAD-SOP to easily determine if flight profiles consist of non-adherence to SOP that could be contributing factors to an accident flight.

6.3.5 FAD-SOP Future work

The framework is in its initial stage of development. It can be further expanded by considering other data sources in its detection phase such as weather data, safety reports, traffic control instructions and cockpit voice recording. Potential technology areas such as voice to text combined with Natural Language Processing are available to mine these voice data which can then provide more insight into why certain decisions may have been made by the flight crew.

Another opportunity for extending the work is to consider other flight phases such as take-off which have a reference point for ensuring that all flight data are comparable. Future work could also evidence the generic capability of the framework for use on other routes, operator, and aircraft types.

The framework was designed with flight safety in mind. It would be useful to ascertain how it performs in other safety domains. The architecture of the framework, however, does provide a robust structure that caters for pre-processing raw data, labelling instances (where majority of the domain data is unlabelled), determining optimal thresholds for individual detectors, combining individual outcomes into an ensemble, and eventually applying reasoning to isolating desired anomaly types.

It would be useful to extend this work by introducing a visual interface capable of showing the similarities between the query and similar case instances thereby

explaining why the SOP-CBR block selected the historic cases it matched with the new query.

6.3.6 FAD-SOP Procedure deviation categories

The research revealed several techniques used by pilots that did not follow the expected procedure and/or basic tenets of flying commercial jet planes like the A320 used in this work. Some of these are outlined briefly below:

Late use of Speed brakes: Speed brakes are commanded later in the approach if there is excess energy to dissipate. They automatically stow when Flap FULL is selected. Late use of speed brakes might indicate poor energy management. In this data, 6 out of the 10,152 flights used the speed brakes.

Autopilots not in same mode: The A320 aircraft possesses two autopilot buttons (one for the captain and the other for the first officer) majorly to aid redundancy. SOPs state that both autopilots should be engaged when the APPR pushbutton is pressed to arm the localizers and Ground Speed modes. Several flights were found to have entered the 90-second window with autopilots in different modes showing that some pilots may have forgotten to do this.

'InLanding3' Configuration not set: The state of the 'InLanding3' feature provides the aircraft with the information that the crew intend to land with low flaps (FLAP 3) configuration. The Flight Crew Operating Manual states: *FLAP FULL will be the landing configuration under normal circumstances. The use of FLAP 3 is recommended if an encounter with windshear / severe turbulence is considered possible on the approach.* Therefore, situations could arise where flap full was not selected due to turbulence. Flap 3 landing are also carried out in abnormal situations, such as technical failures. There were 16 flights that landed with this low flap setting but only the crew on 14 of these flights remembered to 'tell' the aircraft of their intention by switching the 'InLanding3' mode to the ON state. In other words, the flight crew on 2 of the 16 flights failed to follow the guidance in the operating manual.

Nose Gear delay at touchdown: Chapter 4 and 5 provide more details about this technique which was common to several flights in this dataset. On these flights, the

nose gear was held off for longer than usual. Ideally flights would be flown with little variation and within tight tolerances therefore, for several flights having this occurrence within this dataset, a deviation (albeit a low risk one) was concluded. This is because it would prompt further investigations into the previous behaviour of the flight crew to establish whether a technique issue is apparent.

Manually flown flights: Further investigations into some detected anomalous flights revealed that they were manually flown throughout the duration of the flight. In normal circumstances, pilots have guidance to use the automatic systems for certain phases of flight such as cruise. However, some airlines carry out manual flying for training purposes therefore, in this case an expert would carry out no further actions on such flights. However, for a commercial flight, it would be considered an interesting behaviour to investigate further.

High Sink rate: Some flights were observed with interesting high sink rate events which according to the expert “would not have been picked up at the time”. One of these cases showed the aircraft was set up for an Autoland but the captain intervened during flare (the period just before touchdown). The intervention was necessary due to a high sink rate noticed on further investigation of the flight profile.

Late landing configuration: A very high number of flights were observed to have switch from autoland to manual land late into the landing window. This may have been due to unstabilised approaches which transferred unto the landing phase. Many of these seem to have landed with minimal incidents but it is still regarded as unsafe practise or poor technique that would be a good fit for training focus.

6.4 Conclusion

This chapter provides a summary of the outputs from previous chapters and describes how they contributed to the achievement of the overall aim of the research. It highlights the novelty of this research and further provides recommendations for further studies. The chapter also presents the implication and significance of the study to stakeholders

and the industry at large. Some flights of interest were also outlined in this work providing some insights into the varied types of deviations detected by FAD-SOP. Specifically, this research focuses on enhancing the safety levels by identifying more anomalous flight events particularly procedure deviations which is a major source of accidents.

This research confirms the findings of [10] with respect to the difference in the anomalous events that individual flight AD methods are able to detect. It then contributes to existing knowledge related to developing ensembles by focusing on combining methods in a way that identifies a wider array of anomalies in flight data analysis. Therefore, it sought to build on the work of [13] which describes a generic knowledge discovery pipeline or framework for revealing risky events in aviation data while making use of data mining algorithms. FAD-SOP, similarly, uses multiple existing detection methods but extends the work in [13] by introducing an ensemble combination of their output labels and then employing a reasoning system to tailor the framework for isolating events that are solely due to procedure deviation. It is envisaged that this framework is also adaptable to other flight phases as well as tailored for highlighting other specific variants of anomalies for which the user may be concerned.

References

1. Sumwalt, R.L. *Standard Operating Procedures: The backbone of professional flight operations*. 2013 [cited 2020 12.12.2020]; Available from: https://ders.es/Sumwalt_130916_sop.pdf.
2. Foundation, F.S. *Position Paper: Pilot training and competency*. 2018.
3. Nanduri, A. and L. Sherry. *Anomaly detection in aircraft data using Recurrent Neural Networks (RNN)*. in *2016 Integrated Communications Navigation and Surveillance (ICNS)*. 2016. Ieee.

4. (NASA), N.A.a.S.A., *Multiple Kernel Anomaly Detection (MKAD) sample dataset*. 2013.
5. Budalakoti, S., A.N. Srivastava, and M.E. Otey, *Anomaly detection and diagnosis algorithms for discrete symbol sequences with applications to airline safety*. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), 2008. **39**(1): p. 101-113.
6. Li, L., et al., *Analysis of flight data using clustering techniques for detecting abnormal operations*. Journal of Aerospace information systems, 2015. **12**(9): p. 587-598.
7. Li, L., et al. *Anomaly detection in onboard-recorded flight data using cluster analysis*. in *2011 IEEE/AIAA 30th Digital Avionics Systems Conference*. 2011. IEEE.
8. Li, L., et al., *Anomaly detection via a Gaussian Mixture Model for flight operation and safety monitoring*. Transportation Research Part C: Emerging Technologies, 2016. **64**: p. 45-57.
9. Vasquez-Morales, G.R., et al., *Explainable prediction of chronic renal disease in the colombian population using neural networks and case-based reasoning*. IEEE Access, 2019. **7**: p. 152900-152910.
10. Das, S., et al. *Comparison of algorithms for anomaly detection in flight recorder data of airline operations*. in *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*. 2012.
11. Althoff, K.-D., et al., *Case-based reasoning for medical decision support tasks: The Inreca approach*. Artificial Intelligence in Medicine, 1998. **12**(1): p. 25-41.
12. Barry, D.J., *Estimating runway veer-off risk using a bayesian network with flight data*. Transportation Research Part C: Emerging Technologies, 2021. **128**: p. 103180.

13. Matthews, B., et al., *Discovering anomalous aviation safety events using scalable data mining algorithms*. Journal of Aerospace Information Systems, 2014. **11**(7): p. 482-482.

APPENDICES

Appendix 1 – Python code for filtering for the 90-second landing window

```
1 # 90-second window for Landing
2
3 import pandas as pd
4 import glob
5 import numpy as np
6 import matplotlib as plt
7 %matplotlib inline
8 from pathlib import Path
9 import re
10 import statistics
11 from statistics import mean
12 import os
13
14 pd.set_option('display.max_rows', 1000)
15 pd.set_option('display.max_columns', 5000)
16 pd.set_option('display.width', 5000)
17
18 #os.chdir('E:\\PythonFlights\\test\\A320\\CleanBHD-LHR')
19 os.chdir('E:\\PythonFlights\\test\\A320\\CleanLHR-CDG')
20 #pL = Path('E:/PythonFlights/test/A320/ProcessedLandingBHD-LHR')
21 pL = Path('E:/PythonFlights/test/A320/ProcessedLandingLHR-CDG')
22 NaN = np.nan
23 #####
24
25 files1 = [f1 for f1 in glob.glob("*/*.csv", recursive=True)]
26 for f1 in files1:
27
28     print(f1) #List files with path
29     df1 = pd.read_csv(f1)
30     f = df1.tail(2400)#for Last 40mins.Do this before finding 'onground' so as to exclude rows that match criteria
31     #in earlier phases
32
33     #find point of touchdown
34     onground = (((f['Land_Gear_Squat_Switch_-_LH'].values == 1) & (f['Land_Gear_Squat_Switch_-_Nose'].values == 1))
35                 | ((f['Land_Gear_Squat_Switch_-_RH'].values == 1) & (f['Land_Gear_Squat_Switch_-_Nose'].values == 1))
36                 | ((f['Land_Gear_Squat_Switch_-_LH'].values == 1) &
37                   (f['Land_Gear_Squat_Switch_-_RH'].values == 1))).argmax()
38     if onground == 0:
39         print("The row index where", f1, "touches the ground is:", onground) #onground is the index number for
40         #touchdown point
```

```

42 #now select 90 rows for Landing phase (70 <- TD -> 20)
43 landlist = f.iloc[ [onground-69, onground-68, onground-67, onground-66, onground-65, onground-64, onground-63,
44                    onground-62, onground-61, onground-60, onground-59, onground-58, onground-57, onground-56,
45                    onground-55, onground-54, onground-53, onground-52, onground-51, onground-50, onground-49,
46                    onground-48, onground-47, onground-46, onground-45, onground-44, onground-43, onground-42,
47                    onground-41, onground-40, onground-39, onground-38, onground-37, onground-36, onground-35,
48                    onground-34, onground-33, onground-32, onground-31, onground-30, onground-29, onground-28,
49                    onground-27, onground-26, onground-25, onground-24, onground-23, onground-22, onground-21,
50                    onground-20, onground-19, onground-18, onground-17, onground-16, onground-15, onground-14,
51                    onground-13, onground-12, onground-11, onground-10, onground-9, onground-8, onground-7,
52                    onground-6, onground-5, onground-4, onground-3, onground-2, onground-1, onground, onground+1,
53                    onground+2, onground+3, onground+4, onground+5, onground+6, onground+7, onground+8, onground+9,
54                    onground+10, onground+11, onground+12, onground+13, onground+14, onground+15, onground+16,
55                    onground+17, onground+18, onground+19, onground+20] ]
56
57 #Landlist = Landlist.values
58 #print("The selection of rows for Landing based on touchdown pt is:", Landlist)
59
60 #drop average column
61 f = landlist.drop(['AltAvg'], axis = 1)
62
63 #or count the sum of nans in each flight
64 if f.isna().values.sum() != 0:
65     print(f1, f.isna().values.sum())
66
67 #output to file
68 file_name = '{0}'.format(f1)
69 f.to_csv(Path(pL, file_name), index=False)
70
71 #display(f)#new dataframe
72 #dfList2[0]

```

Appendix II – Python Code for Creating Reduced dataset for CAD-Cat detector.

```
1  #Create Reduced categorical dataset for CAD-Cat detector
2  #Create Reduced Continuous dataset for CAD-Cont detector
3
4  df1 = pd.DataFrame(index = range(0,1))
5
6  os.chdir('C:/Users/s268555/OneDrive - Cranfield University/s268555/Working_Data/MKAD_ForSupervisor/Ready33')#RDP SCHPC
7
8  dflistCat = []#dataframe for categorical data type
9
10 dflistCont2 = []#dataframe for continuous data type
11
12 #after importing the original cont n cat values, then use PLA or PAA to reduce them
13 dflistpaa = []
14 dflistypaa = []
15
16 #merged list of dfs
17 dflist2 = []
18 dfraw = []
19
20 window_size = 9 #3, 6, 9, 18
21 paa = PiecewiseAggregateApproximation(window_size=window_size)
22
23 #extract flight filenames
24 filesD = [f for f in glob.glob("*/*.csv", recursive=True)]
25 for f in filesD:
26     print(f)
27     df = pd.read_csv(f)
28
29 #categorical data type segregation
30 dfCat = df[['AP_1_Engaged', 'AutoP2Eng', 'EngagedATS', 'Flap_Angle',
31            'GRight', 'GNose', 'GLeft',
32            'Brake_OnOffAutomatic', 'Reverser_Deployed_Eng_1', 'RevDepldEng2',
33            'Speed_Brake_Command', 'RSpoilOut', 'x10OutSpoilrL', 'GndSpoiler_Armed', 'InLanding_Config_Conf_3',
34            'ATSActive']]
35
```

```

36 #conditions for populating new Cat dataframe
37 #first is the index of the first occurrence of the max value. e.g for flap, many were 2 throughout while some
38 #started with      #1 and then swtched to 2 at some point. a few were 1 throughout. so, for each flight, the
39 #max value will be picked and its indx written down.
40 |
41 row1_col1 = dfCat.AP_1_Engaged.idxmax()#index of first row matching the condition
42 row1_col2 = dfCat.AutoP2Eng.idxmax()#index of first row matching the condition
43 row1_col3 = dfCat.EngagedATS.idxmax()#index of first row matching the condition
44 row1_col4 = dfCat.FlaponAngle.idxmax()#index of first row matching the condition
45 row1_col5 = dfCat.GRight.idxmax()#index of first row matching the condition
46 row1_col6 = dfCat.GNose.idxmax()#index of first row matching the condition
47 row1_col7 = dfCat.GLeft.idxmax()#index of first row matching the condition
48 row1_col8 = dfCat.Brake_OnOffAutomatic.idxmax()#index of first row matching the condition
49 row1_col9 = dfCat.Reverser_Deployed_Eng_1.idxmax()#index of first row matching the condition
50 row1_col10 = dfCat.RevDepldEng2.idxmax()#index of first row matching the condition
51 row1_col11 = dfCat.Speed_Brake_Command.idxmax()#index of first row matching the condition
52 row1_col12 = dfCat.RSpoilOut.idxmax()#index of first row matching the condition
53 row1_col13 = dfCat.x1OutSpoilrL.idxmax()#index of first row matching the condition
54 row1_col14 = dfCat.GndSpoiler_Armed.idxmax()#index of first row matching the condition
55 row1_col15 = dfCat.InLanding_Config_Conf_3.idxmax()#index of first row matching the condition
56 row1_col16 = dfCat.ATSActive.idxmax()#index of first row matching the condition
57
58
59
60 #count how Long it stayed at a particular condition
61 row2_col1 = sum(1 for i in dfCat.AP_1_Engaged if i == 1)
62 row2_col2 = sum(1 for i in dfCat.AutoP2Eng if i == 1)
63 row2_col3 = sum(1 for i in dfCat.EngagedATS if i == 1)
64 row2_col4 = sum(1 for i in dfCat.FlaponAngle if i != 0)
65 row2_col5 = sum(1 for i in dfCat.GRight if i != 0)
66 row2_col6 = sum(1 for i in dfCat.GNose if i != 0)
67 row2_col7 = sum(1 for i in dfCat.GLeft if i != 0)
68 row2_col8 = sum(1 for i in dfCat.Brake_OnOffAutomatic if i == 1)
69 row2_col9 = sum(1 for i in dfCat.Reverser_Deployed_Eng_1 if i == 1)
70 row2_col10 = sum(1 for i in dfCat.RevDepldEng2 if i == 1)
71 row2_col11 = sum(1 for i in dfCat.Speed_Brake_Command if i == 1)
72 row2_col12 = sum(1 for i in dfCat.RSpoilOut if i == 1)
73 row2_col13 = sum(1 for i in dfCat.x1OutSpoilrL if i == 1)
74 row2_col14 = sum(1 for i in dfCat.GndSpoiler_Armed if i == 1)
75 row2_col15 = sum(1 for i in dfCat.InLanding_Config_Conf_3 if i == 1)
76 row2_col16 = sum(1 for i in dfCat.ATSActive if i == 1)

```

```

77
78 #how many times values changed in each column
79 row3_col1 = (dfCat.AP_1_Engaged.fillna(0).diff() != 0).sum()#check if a value has changed
80 row3_col2 = (dfCat.AutoP2Eng.fillna(0).diff() != 0).sum()#check if a value has changed
81 row3_col3 = (dfCat.EngagedATS.fillna(0).diff() != 0).sum()#check if a value has changed
82 row3_col4 = (dfCat.Flap_Angle.fillna(0).diff() != 0).sum()#check if a value has changed
83 row3_col5 = (dfCat.GRight.fillna(0).diff() != 0).sum()#check if a value has changed
84 row3_col6 = (dfCat.GNose.fillna(0).diff() != 0).sum()#check if a value has changed
85 row3_col7 = (dfCat.GLeft.fillna(0).diff() != 0).sum()#check if a value has changed
86 row3_col8 = (dfCat.Brake_OnOffAutomatic.fillna(0).diff() != 0).sum()#check if a value has changed
87 row3_col9 = (dfCat.Reverser_Deployed_Eng_1.fillna(0).diff() != 0).sum()#check if a value has changed
88 row3_col10 = (dfCat.RevDepldEng2.fillna(0).diff() != 0).sum()#check if a value has changed
89 row3_col11 = (dfCat.Speed_Brake_Command.fillna(0).diff() != 0).sum()#check if a value has changed
90 row3_col12 = (dfCat.RSpoilOut.fillna(0).diff() != 0).sum()#check if a value has changed
91 row3_col13 = (dfCat.x1OutSpoilrL.fillna(0).diff() != 0).sum()#check if a value has changed
92 row3_col14 = (dfCat.GndSpoiler_Armed.fillna(0).diff() != 0).sum()#check if a value has changed
93 row3_col15 = (dfCat.InLanding_Config_Conf_3.fillna(0).diff() != 0).sum()#check if a value has changed
94 row3_col16 = (dfCat.ATSActive.fillna(0).diff() != 0).sum()#check if a value has changed
95
96
97 df1[['AP_1_Engaged', 'AutoP2Eng', 'EngagedATS', 'Flap_Angle',
98      'GRight', 'GNose', 'GLeft',
99      'Brake_OnOffAutomatic', 'Reverser_Deployed_Eng_1', 'RevDepldEng2',
100     'Speed_Brake_Command', 'RSpoilOut', 'x1OutSpoilrL', 'GndSpoiler_Armed', 'InLanding_Config_Conf_3',
101     'ATSActive']] = pd.DataFrame([[row1_col1, row1_col2, row1_col3, row1_col4, row1_col5, row1_col6, row1_col7,
102     row1_col8, row1_col9, row1_col10, row1_col11, row1_col12, row1_col13, row1_col14, row1_col15,
103     row1_col16]], index=df1.index)
104
105 #second row inserted into new df'
106 df1.loc[1] = [row2_col1, row2_col2, row2_col3, row2_col4, row2_col5, row2_col6, row2_col7, row2_col8, row2_col9, row2_col10,
107     row2_col11, row2_col12, row2_col13, row2_col14, row2_col15, row2_col16]
108
109
110 #third row inserted into new df
111 df1.loc[2] = [row3_col1, row3_col2, row3_col3, row3_col4, row3_col5, row3_col6, row3_col7, row3_col8, row3_col9, row3_col10,
112     row3_col11, row3_col12, row3_col13, row3_col14, row3_col15, row3_col16]
113
114 #display(df1)
115
116 #append df for each flight for continous use in notebook
117 df1listCat.append(dfCat)
118
119
120 df1raw.append(df)
121
122

```

Appendix III – Python Code to Create PLA reduced data set for CAD-Cont detector

```
124
125
126 #exponential smoothing for ONLY the 32 CONTINUOUS cols.
127 dfCont2 = df[['Computed_Airspeed', 'Ground_Speed', 'Radio_Altitude_1', 'Roll_Angle', 'Pitch_Angle',
128              'Throttle_Angle_Eng_1', 'Slat_Position', 'Approach_Speed_Target',
129              'Normal_Acceleration', 'Longitudinal_Acceleration', 'Lateral_Acceleration',
130              'SSCaptPitch', 'CptRollSS', 'FOPitchSS', 'Sidestick_Pos_FO_Roll', 'Wind_Speed',
131              'VMAX_Operating_Airspeed', 'PosRH5Spoiler', 'RH3Spoiler', 'SpoilerPos_RH_2', 'LH5Spoiler',
132              'SplerLH4', 'x2LHSpoilr', 'Rudder_Position', 'RgtAileron', 'All_Speed_Aileron_Pos_Left', 'SlctdVS',
133              'TrackSel', 'HeadingSel', 'FltPathSel', 'DecnHeightSel',
134              'AltiSel']].transform(lambda x: x.ewm(alpha=0.33, adjust=False).mean())
135
136 dflistCont2.append(dfCont2)
137
138 #covert dfCont to array for merge with dfcat, and perform PLA on it for dimensionality reduction
139 values1 = dfCont2.to_numpy().T
140 X_paa = paa.transform(values1)
141
142 dfX_paa = pd.DataFrame(X_paa.T,
143                        columns = ['Computed_Airspeed', 'Ground_Speed', 'Radio_Altitude_1', 'Roll_Angle', 'Pitch_Angle',
144                                'Throttle_Angle_Eng_1', 'Slat_Position', 'Approach_Speed_Target',
145                                'Normal_Acceleration', 'Longitudinal_Acceleration', 'Lateral_Acceleration',
146                                'SSCaptPitch', 'CptRollSS', 'FOPitchSS', 'Sidestick_Pos_FO_Roll', 'Wind_Speed', 'VMAX_Operating_Airspeed',
147                                'PosRH5Spoiler', 'RH3Spoiler', 'SpoilerPos_RH_2', 'LH5Spoiler', 'SplerLH4', 'x2LHSpoilr', 'Rudder_Position',
148                                'RgtAileron', 'All_Speed_Aileron_Pos_Left', 'SlctdVS', 'TrackSel', 'HeadingSel',
149                                'FltPathSel', 'DecnHeightSel', 'AltiSel'])
150
151 dflistpaa.append(dfX_paa)
152
153
```

Appendix IV – Median-based Euclidean distance Measure for one Parameter (Radio_Altitude)

```

1  #Find the Median Profile of the 10,152 flights|
2
3  '''
4  fCat = f[['AP_1_Engaged', 'AutoP2Eng', 'EngagedATS', 'Flap_Angle',
5           'GRight', 'GNose', 'GLeft',
6           'Brake_OnOffAutomatic', 'Reverser_Deployed_Eng_1', 'RevDepldEng2',
7           'Speed_Brake_Command', 'RSpoilOut', 'x1OutSpoilR', 'GndSpoiler_Armed', 'InLanding_Config_Conf_3',
8           'ATSActive']]
9
10 fCont=
11 'Computed_Airspeed', 'Ground_Speed', 'Radio_Altitude_1', 'Roll_Angle', 'Pitch_Angle', 'Throttle_Angle_Eng_1',
12 'Slat_Position', 'Approach_Speed_Target', 'Normal_Acceleration',
13 'Longitudinal_Acceleration', 'Lateral_Acceleration', 'SSCaptPitch', 'CptRollSS', 'FOPitchSS', 'Sidestick_Pos_FO_Roll',
14 'Wind_Speed', 'VMAX_Operating_Airspeed',
15 'PosRH5Spoiler', 'RH3Spoiler', 'SpoilerPos_RH_2', 'LH5Spoiler', 'SplrLH4', 'x2LH5Spoilr',
16 'Rudder_Position', 'RgtAileron', 'All_Speed_Aileron_Pos_Left', 'SlctdVS', 'TrackSel', 'HeadingSel', 'FltPathSel',
17 'DecnHeightSel', 'AltiSel'
18 '''
19
20 from scipy.spatial import distance #for euclidean
21
22 Euclist = []# List of eucl distances from median
23 colist = []#List of specific parameter values for all the flights
24
25 for df, f in zip(dfrw, filesD):
26     #for column in df:
27     colval = df['Radio_Altitude_1']
28     colist.append(colval)#List of all 90secs for all 10k flts
29
30     #to view shape and perform median, convert to df
31     colDF = pd.DataFrame (colist)
32
33     #mean flight of the 10k flight with its 90sec timestamps in 1-d array
34     colMed = colDF.median()

```

```

1  #Run on 10k flights to determine how different they are from the median profile determined in prev cell
2  #then save the distances and fileIDs to text file
3
4  flt = open("C:/Users/s268555/OneDrive - Cranfield University/s268555/Working_Data/Fltlist4k.txt", "w")
5  dist = open("C:/Users/s268555/OneDrive - Cranfield University/s268555/Working_Data/distlist4k.txt", "w")
6
7  outlistParam = []#outliers specific to individual params
8  colist2 = []#List of GroundSpeed values for all 10k flights
9  Euclist2 = []# List of eucl distances from median for all 10k flights
10 outdfs = []#List of all outlier dataframes
11
12 #heterogeneous data for all flights similiarity measure to median profile
13 for df, f in zip(dfrw, filesD):#change to dflist2 and files1 for all 10k flights
14     #for df, f in zip(dflist2, files1):
15     colval2 = df['Radio_Altitude_1']
16     colist2.append(colval2)
17
18     #calculate euclidean distance between df[Groundspeed] 1-D array and colMed 1-D array
19     colEuc2 = distance.euclidean(colval2, colMed)#euclidean adds all points and finds the sqrt of the sum.
20     Euclist2.append(colEuc2)
21 #Euclist2
22
23 #find flights with distance > threshold distance
24 #Add chosen items to result List
25 if colEuc2 > 400 and colEuc2 <= 750:
26     #if colEuc2 >= 500: #CHANGE THE VALUE DEPENDING ON PARAMETER THRESHOLD ANALYSIS IN PREV CELL
27     #print("outlier flight:", f)
28     outdfs.append(colval2)
29     outlistParam.append(f)
30
31 #write to file
32 flt.write("%s\n" %f)
33 strlbl = str(colEuc2)
34 dist.write("%s\n" %strlbl)
35 dist.close()
36 flt.close()
37
38 outlistParam
39

```

Appendix V – Python code for main ClusterAD-DataSample Implementation

```
1  #!/usr/bin/env python
2
3  import numpy as np
4  import pandas as pd
5
6  import sys
7
8  from tqdm import tqdm
9  from copy import deepcopy
10
11 from sklearn.mixture import GaussianMixture
12 from numpy.linalg import norm
13 from copy import deepcopy
14
15
16 class GaussianMixtureInTimeAnomalyDetector:
17     """
18     ClusterAD-DataSample anomaly-detection method implementation
19     """
20     def __init__(self,
21                 n_components=None,
22                 n_min=1,
23                 n_max=10,
24                 step=1,
25                 tol=1e-6,
26                 covariance_type='diag',
27                 init_params='kmeans',
28                 random_state=None,
29                 verbose=0,
30             ):
31         """
32         Constructor accepts args for sklearn.mixture.GaussianMixture inside.
33         Default params are choosen as the most appropriate
34         for flight-anomaly-detection problem
35         according the original article.
36         """
37         assert n_components is not None or (n_min is not None and
38                                           n_max is not None)
39
40         self.n_min = n_min
41         self.n_max = n_max
42         self.step = step if step is not None else 1
43         self.n_components = n_components
44         self.tol = tol
45         self.covariance_type = covariance_type
46         self.init_params = init_params
47         self.random_state = random_state
```

```

48     self.verbose = verbose
49
50     self.eps = 1e-12 # feature-normalization constant
51
52     def fit(self, X):
53         """
54         X must contains F objects time series with
55         length T vectors-features with size N each
56         i. e. X.shape is (F, N, M)
57         """
58
59         X = np.array(X)
60
61         assert len(X.shape) == 3
62         self.F, self.T, self.N = X.shape
63
64         # prepare data for fitting
65         X = X.reshape(self.F * self.T, self.N)
66
67         self.data_mean = np.mean(X, axis=0)
68         self.data_std = np.std(X, axis=0) + self.eps
69
70         X = self._normalize(X)
71
72         if self.n_components is None:
73             self.bics = []
74             gms = []
75             n_range = np.arange(self.n_min, self.n_max + 1, self.step)
76             for n in tqdm(n_range, position=0, file=sys.stderr):
77                 gm = GaussianMixture(
78                     n_components=n,
79                     tol=self.tol,
80                     covariance_type=self.covariance_type,
81                     init_params=self.init_params,
82                     random_state=self.random_state,
83                 )
84                 try:
85                     gm.fit(X)
86                     self.bics.append(gm.bic(X))
87                     gms.append(gm)
88                 except:
89                     self.bics.append(np.inf)
90                     gms.append(None)

```

```

91
92
93     best_bic = np.min(self.bics)
94     best_n = self.n_min + np.argmax(self.bics) * self.step
95     gm = gms[np.argmax(self.bics)]
96
97     if gm is None:
98         print('Bad n_components number is choosen')
99         return
100
101     self.n_components = best_n
102
103     print('Best number of clusters (according to BIC) - {}'.format( \
104         self.n_components),file=sys.stderr)
105
106     else:
107         gm = GaussianMixture(
108             n_components=self.n_components,
109             tol=self.tol,
110             covariance_type=self.covariance_type,
111             init_params=self.init_params,
112             random_state=self.random_state,
113             )
114
115         gm.fit(X)
116
117     print('Sample clusters fitting - OK.',file=sys.stderr)
118
119     self.X = X.reshape(self.F, self.T, self.N)
120
121     self.cluster_weights = gm.weights_
122     self.cluster_means = gm.means_
123     self.cluster_covariances = gm.covariances_
124
125     self.__memorize_probs()
126
127     print('Probabilities memorization - OK.',file=sys.stderr)
128
129     likelihood = self.__evaluate_log_likelihood(self.X)
130
131     print('Log-likelihood evaluation - OK.',file=sys.stderr)
132
133     return likelihood

```

```

135 def predict(self, X):
136     ...
137     Calculate log-likelihood for each one-time-slice sample,
138     dim(X) = 3, X.shape[1] must be equal time length(shape[1])
139     Normalize samples by train data and call evaluation log likelihood method
140     ...
141     X = np.array(X)
142     assert len(X.shape) == 3
143     self.F, self.T, self.N = X.shape
144     assert X.shape[1] == self.T
145     X = self._normalize(X)
146
147     return self.__evaluate_log_likelihood(X)
148
149
150 def predict_point(self, X):
151     ...
152     Calculate log-likelihood for single sample (not series or group of series)
153     ...
154     pass
155
156 def __memorize_probs(self):
157     # memorization all P(cluster/sample)
158     self.__p_cluster_sample = np.zeros((self.n_components, self.T, self.F))
159
160     for series in tqdm(np.arange(self.F), position=0):
161         for time in np.arange(self.T):
162             probs = [self.cluster_weights[i] * self.__p_sample_cluster(self.X[series][time], i) \
163                     for i in np.arange(self.n_components)]
164             norma = np.sum(probs)
165
166             for cluster in np.arange(self.n_components):
167                 self.__p_cluster_sample[cluster][time][series] = probs[cluster] / (norma + self.eps)
168
169     # memorization all P(cluster/time)
170     self.__p_cluster_time = np.zeros((self.n_components, self.T))
171
172     for time in np.arange(self.T):
173         for cluster in np.arange(self.n_components):
174             self.__p_cluster_time[cluster][time] = self.__get_p_cluster_time(cluster, time)
175

```

```

177 def _normalize(self, X):
178     return (X - self.data_mean) / self.data_std
179
180 def __diag_gauss_pdf(self, x, mean, cov):
181     """
182     Custom calculation gaussian density in case if covariance is diagonal matrix
183     cov is array of covariance matrix diagonal elements
184     """
185     delta = np.array(x) - np.array(mean)
186     #print(delta)
187     #print((np.array(cov) + self.eps))
188     inv = 1 / (np.array(cov) + self.eps)
189     logp = -0.5 * ((delta.dot(inv * delta)) + np.log(np.prod(cov) + self.eps) + self.N * np.log(2 * np.pi))
190
191     return np.exp(logp)
192
193
194 def __p_sample_cluster(self, x, cluster):
195     """
196     Conditional likelihood(sample|cluster)
197     """
198     return self.__diag_gauss_pdf(x,
199                                 self.cluster_means[cluster],
200                                 self.cluster_covariances[cluster])
201
202
203 def __get_p_cluster_time(self, cluster, t):
204     clusters_probs = [np.sum((self.__p_cluster_sample[i][t])) for i in np.arange(self.n_components)]
205     #print(clusters_probs)
206     return clusters_probs[cluster] / np.sum(clusters_probs)
207
208
209 def __evaluate_sample_in_time(self, x, t):
210     """
211     Evaluation in population log likelihood for time-slice normalized sample x_t.
212     x is M-dimentional one-time-slice sample
213     t is order(time) of sample x in time series
214     t must be in [0, 1, ... N)
215     """
216     #print(np.sum([self.__p_sample_cluster(x, cluster) * self.__p_cluster_time[cluster][t] \
217                  #for cluster in np.arange(self.n_components)]))
218
219     return np.log(np.sum([self.__p_sample_cluster(x, cluster) * self.__p_cluster_time[cluster][t] \
220                          for cluster in np.arange(self.n_components)])) #added the small amount to stop the 'divide by ze

```

```

222
223 def __evaluate_log_likelihood(self, X):
224     log_likelihood = np.zeros(X.shape[:2])
225     for f in np.arange(X.shape[0]):
226         for t in np.arange(X.shape[1]):
227             #X[X==0] = 0.00001
228             log_likelihood[f][t] = self.__evaluate_sample_in_time(X[f][t], t)
229
230     return log_likelihood
231
232
233 def find_anomalies(self, scores, strategy='sample', anomaly_top=0.01, log_likelihood_threshold=None):
234     '''
235     Extract abnormal samples.
236     Args:
237         scores - log_likelihoods for each one-time-slice sample,
238         dim(X) = 2, X.shape[1] must be equal time length(shape[1])
239         strategy(default: 'sample') - way of results representation:
240             'sample' - find abnormal one-time-slice samples
241             'series' - find abnormal series (anomaly of each series is sum of samples log likelihoods)
242         log_likelihood_threshold is used as anomaly-data upper bound if it's specified, else
243         anomaly_top will be used and converted to population log_likelihood_threshold
244     Return:
245         sorted list of anomalies - [(log_likelihood, (series, [time])),]
246         log_likelihood_threshold - upper bound of abnormal samples
247     '''
248     scores = np.array(scores)
249
250     assert len(scores.shape) == 2
251     assert scores.shape[1] == self.T
252     assert anomaly_top > 0 or log_likelihood_threshold is not None
253
254     if strategy == 'sample':
255         serialized_scores = [(scores[series][time], (series, time))
256                               for series in np.arange(scores.shape[0]) for time in np.arange(scores.shape[1])]
257
258     elif strategy == 'series':
259         serialized_scores = [(np.sum(scores[series]), (series))
260                               for series in np.arange(scores.shape[0])]
261
262     else:
263         raise ValueError("strategy must be in {}".format(['sample', 'series']))

```

```

263     serialized_scores.sort()
264
265     sorted_scores = np.array([s[0] for s in serialized_scores])
266     size = len(sorted_scores)
267
268     if log_likelihood_threshold is None:
269         bound = int(size * anomaly_top)
270         bound = min(size, bound + 1)
271         log_likelihood_threshold = sorted_scores[max(0, bound - 1)]
272     else:
273         bound = np.argmax(sorted_scores > log_likelihood_threshold)
274         if bound == 0 and sorted_scores[0] <= log_likelihood_threshold:
275             bound = len(sorted_scores)
276
277     return serialized_scores[:bound], log_likelihood_threshold
278
279
280
281 ### Additional functionality for specific tasks ###
282
283
284
285 def smoothed_sample_anomalies(scores, halflife=2):
286     """
287     Exponential weighted sample likelihood.
288     Args:
289         scores - result of running ClusterAD-DataSample algorithm for samples
290         halflife - size of smoothing window.
291     """
292     frames = [pd.DataFrame(series) for series in scores]
293     return np.array([np.array(series.ewm(halflife).mean()).reshape(-1) for series in frames])
294
295
296 def extract_anomaly_target(frame, frame_period, halflife,
297                             horizon, top=0.01, more_info=False, **kwargs):
298     """
299     Extract target anomalies from 2-dim frame.
300     Args:
301         frame - 2-dim pandas Dataframe
302         frame_period - period for division into group of independent series
303         halflife - size of smoothing window
304         top - anomaly bound
305         more_info - return more important objects if True
306     """

```

```

306     '''
307     assert len(frame.shape) == 2
308     assert isinstance(frame, pd.DataFrame)
309     frame = np.array(deepcopy(frame))
310     start_size = frame.shape[0]
311     if frame.shape[0] % frame_period != 0:
312         print("remove last elements until period", file=sys.stderr)
313         for _ in np.arange(frame.shape[0] % frame_period):
314             frame = np.delete(frame, -1, 0)
315
316     data = frame.reshape(-1, frame_period, frame.shape[1])
317     detector = GaussianMixtureInTimeAnomalyDetector(**kwargs)
318     scores = detector.fit(data)
319     smoothed_scores = detector.smoothed_sample_anomalies(scores, halflife)
320     anomalies, treshold = detector.find_anomalies(smoothed_scores, anomaly_top=top)
321     anomaly_indexes = [t[1][0] * frame_period + t[1][1] for t in anomalies]
322     all_anomalies = set()
323     for a in anomaly_indexes:
324         for l in np.arange(horizont):
325             all_anomalies.add(max(a - l, 0))
326
327     targets = np.zeros(start_size)
328
329     for a in all_anomalies:
330         targets[a] = 1
331
332     if more_info:
333         return targets, scores, smoothed_scores, treshold
334     else:
335         return targets

```

Appendix VI – MATLAB code for MKAD detector

```
1
2
3 disp('Creating Training Kernels')
4 templ = cputime;
5 [Kd,Kc]=generateKcKd_FromPreprocessed(SAXFilesDir,trlist,trlist);
6 fprintf('\n\n');
7
8
9 %% Generate Composite Kernel (Heterogeneous algo)
10
11 % combine kernels
12 K = betac*Kc + betad*Kd;
13
14 % build one-class SVMs model
15 [model] = build_ocmodel(K,nu);
16
17 % Lagrangian multipliers for support vectors
18 alpha = model.alpha;
19 % Bias term
20 rho = model.rho;
21 % position of support vectors
22 pos = model.positions;
23 % bias from contineous data
24 rhoc = -(1/length(pos))*sum(alpha'*0.5*Kc(pos,pos));
25 % bias from discrete data
26 rhod = -(1/length(pos))*sum(alpha'*0.5*Kd(pos,pos));
27 model.rhoc = rhoc;
28 model.rhod = rhod;
29
30 %keep nu betac betad x_opti verbose str_ker_rbf use_conti_rbf kyle...
31 % model alpha rho pos type Distrib delay no_of_flight_examples idex half_length_of_each_flight
32 clear K Kc Kd p_A p_B tmpk
33
```

```

1 % Generate SAX Files
2 % If reference list is present load from file.
3 if(haveRef)
4     dataref = refFile;
5
6     if(~exist(dataref,'file'))
7         error(['Could not find ',dataref,'. Please check the file name and make sure it is in Filelists/ directory ']);
8     end
9     dataref = importdata(dataref);
10    if(exist('testFile','var'))
11        if(exist(testFile,'file'))
12            allFileList = importdata(testFile);
13        else
14            fprintf('No test file specified using all files in directory for testing...\n');
15            allFileList = dir([SRCpath,'*.csv']);
16            tmp=struct2cell(allFileList);
17            allFileList=tmp(1,:);
18        end
19    else
20        fprintf('No test file specified using all files in directory for testing...\n');
21        allFileList = dir([SRCpath,'*.csv']);
22        tmp=struct2cell(allFileList);
23        allFileList=tmp(1,:);
24    end
25 else
26     if(exist('testFile','var'))
27         allFileList = dir([SRCpath,'*.csv']);
28         tmp=struct2cell(allFileList);
29         allFileList=tmp(1,:);
30         rp = randperm(length(allFileList));
31         dataref = allFileList(rp(1:min(50,floor(0.10*length(rp)))));
32         if(exist(testFile,'file'))
33             allFileList=importdata(testFile);
34         else
35             error(['Could not find ',testFile,'. Please check the file name and make sure it is in Filelists/ directory ']);
36         end
37     else
38         allFileList = dir([SRCpath,'*.csv']);
39         tmp=struct2cell(allFileList);
40         allFileList=tmp(1,:);
41         rp = randperm(length(allFileList));
42         dataref = allFileList(rp(1:min(50,floor(0.10*length(rp)))));
43     end
44 end

```

```

46
47 if(haveRef)
48     clear trlist tstlist alllist FilesNotFoundList;
49     tmpIdx=1;
50     tmpIdx2=1;
51     notFoundFlag=0;
52     for i=1:length(dataref)
53         if(exist([SRCpath,dataref(i),'file'])
54             trlist(tmpIdx).name=cell2mat(dataref(i));
55             alllist(tmpIdx).name=cell2mat(dataref(i));
56             tmpIdx=tmpIdx+1;
57         else
58             FilesNotFoundList.trlist(tmpIdx2)=dataref(i);
59             tmpIdx2=tmpIdx2+1;
60             notFoundFlag=1;
61         end
62     end
63     if(~exist('trlist','var'))
64         error('No files from the training list were found in the Source Directory. Please check file list');
65     end
66     tmpIdx2=1;
67     tmpIdx3=1;
68     for i=1:length(allFileList)
69         if(exist([SRCpath,allFileList(i),'file'])
70             alllist(tmpIdx).name=cell2mat(allFileList(i));
71             tstlist(tmpIdx3).name=cell2mat(allFileList(i));
72             tmpIdx=tmpIdx+1;
73             tmpIdx3=tmpIdx3+1;
74         else
75             FilesNotFoundList.trlist(tmpIdx2)=allFileList(i);
76             tmpIdx2=tmpIdx2+1;
77             notFoundFlag=1;
78         end
79     end
80     if(~exist('alllist','var'))
81         error('No files from the full list were found in the Source Directory. Please check file list');
82     end
83
84     if(notFoundFlag)
85         fprintf('Some files were not found in the %s or %s list.\nPlease check your list. Missing files can be found in variable FilesNotFoundList\nContinuing with remaining list...\n',dataref,allFileList);
86     end
87 end

```

```

88- else
89-     [dataref]=FindMostNominalFlights(SRCpath)';
90-     if (~exist('testFile','var'))
91-         datatst=dir([SRCpath,'/*.csv']);
92-         for i=1:length(datatst)
93-             alllist(i).name=datatst(i).name;
94-         end
95-         tstlist = alllist;
96-     else
97-         alllist=cell2struct(allFileList,'name',length(allFileList));
98-     end
99-     for i=1:length(dataref)
100-         trlist(i).name=cell2mat(dataref(i));
101-     end
102-     clear dataref datatst
103- end
104- clear C indx tmp;
105-
106-
107-
108-
109- %% Preprocess all files
110- % zscore normalization of data
111- [DataZscored,Statistics]=generateStat(SRCpath,trlist,channelContinuous);
112- tic
113- h = waitbar(0,'Converting All Files...');
114- fprintf('Running SAX Conversion...\n');
115- for i=1:length(alllist)
116-     SAXFormatFlight(SRCpath,alllist(i).name,SAXFilesDir,Statistics,channelContinuous,channelDiscrete>windowLength,alphabet_size);
117-     waitbar(i / length(alllist));
118- end
119- close(h);
120- pause(0.001);
121- fprintf('SAX conversion done. Elapsed Time: %f Seconds\n',toc);
122-
123-

```

```

51- clc
52- clear all
53- close all
54- fclose('all');
55-
56- %% Adding path of necessary folders and...
57- %% cleaning folders with existing Discrete data files:
58- if(isunix)
59-     addpath('./OtherRoutines');
60-     [SeqPath]=addPath_4unix;
61- else
62-     addpath('./\OtherRoutines');
63-     [SeqPath]=addPath_4win;
64- end
65-
66- if(~exist('Results'))
67-     mkdir('Results');
68- end
69- %%-----
70- %% User input
71- %%-----
72-
73- %%% General Path and File Name Parameters %%%
74- % Source (data) directory
75- SRCpath = 'ReadyMATLAB/';
76- % Working directory
77- DSTpath = 'Results/';
78- % Save the results
79- savefilename=['run_sax_tr_tst_modularCode_0p05nu_', date];
80- % ParamNamesFile='ImportantParameterNames.txt';
81- ParamNamesFile='MATLAB_MKAD/MKAD/ParameterLists/ImportantParameterNames.txt';
82- % Preprocessing Variables
83- SAXFilesDir = 'SAXFiles/Demo/';
84-
85- %%% (Optional) File Lists for Reference and Test %%%
86- refFile = 'MATLAB_MKAD/MKAD/Filelists/RefSet.txt';
87- testFile = 'MATLAB_MKAD/MKAD/Filelists/TestSet.txt';
88-
89- %%% list of continuous and discrete parameters %%%
90- channelContinuous = importdata('MATLAB_MKAD/MKAD/ParameterLists/channelContinuous.txt');
91- channelDiscrete = importdata('MATLAB_MKAD/MKAD/ParameterLists/channelDiscrete.txt');
92-

```

Core MKAD script

```

94  %%% OCSVM & Kernel Parameters %%%
95  %Set model parameters
96  nu=0.05; % nu parameter for 1-class
97  % The weights of the kernels are related as: betac+betad = 1.
98  % The algorithm assumes equal weights of the kernels.
99  % The weights are not optimized in the current version.
100 betac =0.5; % weight of contineous kernel
101 betad =0.5; % weight of discrete kernel
102
103 %%% SAX Parameters %%%
104 % sequence_size = 10; % Choose length of the window (for SAX)
105 % alphabet_size = 10; % Choose an alphabet size (for SAX)
106 windowLength=30; %30 seconds.
107 alphabet_size=10;
108
109
110 %% End User Inputs %%
111 if(exist('refFile','var'))
112     haveRef = 1;
113 else
114     haveRef = 0;
115 end
116
117
118 %%% Step 1. Run Preprocessing step to extract SAX Features and Discrete Sequences %%%
119 generateSAXFiles
120 save([DSTpath, savefilename '.mat'],'trlist','tstlist');
121
122 %%% Step 2. Create Training Kernel from Preprocessed files %%%
123 load([DSTpath, savefilename '.mat']);
124 createTrainingKernel
125 save([DSTpath, savefilename '.mat'],'model','trlist','tstlist'); %Save Training List and Testing List for future runs.
126
127 %%% Step 3. Use Support Vectors defined by Training step to create Test Kernel and
128 %%% Evaluate all flights to find outliers %%%
129 load([DSTpath, savefilename '.mat']);
130 createTestKernel
131 save([DSTpath, savefilename '.mat']);
132 disp('MKAD done...')
133
134 %%% Variable OutlierList Contains the list of anomalous flights identified by MKAD. %%%
135
136 %%% Step 4. (optional) Do Local analysis to determine which parameters were anomalous %%%
137 localAnalysis
138

```

Local Analysis script (post processing)

```

1- C=importdata(ParamNamesFile);
2- if isempty(strmatch('AltitudeChannel',C))
3-     error(['AltitudeChannel not defined in ',ParamNamesFile,'. Please specify']);
4- else
5-     tmp=C{strmatch('AltitudeChannel',C)};
6-     indx=strfind(tmp,'');
7-     AltitudeChannelName=tmp(indx(1)+1:indx(end)-1);
8- end
9-
10- WriteSQMfiles2('ExampleFiles/SQMFile/',SAXFilesDir,trlist,tstlist);
11- HTMLreportV2(SRCpath,DSTpath,Kc,Kd,score,channelContinuous,channelDiscrete,model,trlist,tstlist,SeqPath,savefilename,1,AltitudeChannelName);
12-

```

Appendix VII – MATLAB code for Image Processing Steps

```

1  % Define parameters to use
2  pnames = {'Computed_Airspeed', 'Ground_Speed', 'Radio_Altitude_1',...
3           'Roll_Angle', 'Pitch_Angle', 'Throttle_Angle_Eng_1',...
4           'Slat_Position', 'Approach_Speed_Target',...
5           'Normal_Acceleration', 'Longitudinal_Acceleration',...
6           'Lateral_Acceleration', 'SSCaptPitch', 'CptRollSS', 'FOPitchSS',...
7           'Sidestick_Pos_FO_Roll', 'Wind_Speed', 'VMAX_Operating_Airspeed',...
8           'PosRH5Spoiler', 'RH3Spoiler', 'SpoilerPos_RH_2', 'LH5Spoiler', 'SplerLH4', 'x2LH5Spoilr',...
9           'Rudder_Position', 'RgtAileron', 'All_Speed_Aileron_Pos_Left', 'SlctdVS', 'TrackSel',...
10          'HeadingSel', 'FltPathSel', 'DecnHeightSel', 'AltiSel', 'AP_1_Engaged',...
11          'AutoP2Eng', 'EngagedATS', 'Flap_Angle', 'Park_Brake_OnOff', 'GRight', 'GNose',...
12          'GLeft', 'Brake_OnOffAutomatic', 'Reverser_Deployed_Eng_1',...
13          'Speed_Brake_Command', 'RSpoilOut', 'x1OutSpoilrL', 'GndSpoiler_Armed',...
14          'InLanding_Config_Conf_3', 'ATSActive'};
15
16
17  % Init datastore and tall-array to enable parallel processing of stats
18  % summary
19
20  ds = datastore('Ready33\*.csv');
21  tt = tall(ds);
22  i = 1;
23  tb = readtable(ds.Files{i});
24  for i = 1:length(pnames)
25      rs = contains(tb.Properties.VariableDescriptions, pnames{i});
26      n = find(rs == 1);
27      if ~isempty(n)
28          pnames{i} = tb.Properties.VariableNames{n};
29      end
30  end
31  ds.SelectedVariableNames = pnames;
32  S = summary(tt);
33  for i = 1:length(ds.Files)
34      tb = readtable(ds.Files{i});
35
36      for pj = 1:48
37          pn = S.(pnames{pj});
38          rng = pn.Max - pn.Min;
39          v = (tb.(pnames{pj}) - pn.Min)*255/rng;
40          idta(:,pj) = v;
41      end
42      fn = strrep(ds.Files{i}, 'csv', 'png');
43      imwrite(uint8(idta), fn);
44  end
45

```

Image Processing: Cluster Normal examples

```

22 %% Create image versions of normal examples
23 dsn = datastore('E:\PythonFlights\test\A320\MATLAB_MKAD\ImageProcessing\NormalExamples\*.csv');
24 %% Create image files by scaling numeric values to 0-255 (use max min of entire data param-wise)
25 for i = 1:length(dsn.Files)
26     tb = readtable(dsn.Files{i});
27
28     for pj = 1:48
29         pn = S.(pnames{pj});
30         rng = pn.Max - pn.Min;
31         v = (tb.(pnames{pj}) - pn.Min)*255/rng;
32         idta(:,pj) = v;
33     end
34     fn = strrep(dsn.Files{i}, 'csv', 'png');
35     imwrite(uint8(idta),fn);
36 end
37
38
39 % Now cluster normal examples
40 dtaDir = 'E:\PythonFlights\test\A320\MATLAB_MKAD\ImageProcessing\NormalExamples\*.png';
41 NvidImgs = imageDatastore(dtaDir);
42 NimageIndex = indexImages(NvidImgs);
43 save vi_ClusterImages_RefData

```

Appendix VIII – Python/MATLAB Code for McNemar's Test

Generate Contingency Table

```
# prints the mcnemar table

import numpy as np
from mlxtend.evaluate import mcnemar_table

tb = mcnemar_table(y_target=Tgt,
                   y_model1=CADCat,
                   y_model2=ImgCorr)

print(tb)

[[336  10]
 [  6  16]]
```

Produce Statistic, P-Value and Hypothesis decision (null or alternative)

```
In [50]: # VALIDATION DATASET
# Example of calculating the mcnemar test
from statsmodels.stats.contingency_tables import mcnemar
# define contingency table
table = [[309, 35],
         [14, 30]]
# calculate mcnemar test
result = mcnemar(table, exact=True)
# result = mcnemar(table, exact=True, correction=False) #McNemar's Test with the continuity correction

# summarize the finding
print('statistic=%.3f, p-value=%.3f' % (result.statistic, result.pvalue))
# interpret the p-value
alpha = 0.05
if result.pvalue > alpha:
    # print('Same proportions of errors (fail to reject H0)')
else:
    # print('Different proportions of errors (reject H0)')

statistic=14.000, p-value=0.004
Different proportions of errors (reject H0)
```

MATLAB Code

```
>> n = find(tbl.Tgt == 0);  
>> ntbl = tbl(n,:);  
>> [H,p] = testcholdout(tbl.MKADLb1,tbl.LogOR,tbl.Tgt)
```

H =

logical

1

p =

0.0026