MILA, Université de Montréal

Automated Pilot Performance Assessment

by

Ryan Mokarian

Department of Computer Science and Operations Research

October 2020

Résumé

Ce projet a appliqué une classe d'algorithmes probabiliste d'intelligence artificielle, dans un domaine d'apprentissage non supervisé appelé Gaussian Mixture Model (GMM) pour regrouper les données d'entraînement des pilotes et détecter les vols avec des modèles de données inhabituels. L'approche peut encourager la capacité des outils de formation des pilotes adaptatifs basés sur les données et aider les instructeurs à sélectionner les exercices de formation les plus appropriés, en garantissant des recommandations de formation ciblées.

Pour atteindre l'objectif de recherche, un pipeline allant de l'ensemble de données à l'entrée GMM a été conçu. Ensuite, des modèles de comparaison de vols ont été étudiés et un modèle a été sélectionné. Pour la détection des valeurs aberrantes, une «méthode de calcul de la moyenne des probabilités» a été proposée et expérimentée par rapport à la «méthode de densité log-probabilité» et cette dernière a été sélectionnée. GMM effectué sur les données traitées et les résultats ont montré que GMM peut identifier les grappes et détecter les anomalies parmi les vols au niveau des manœuvres et des événements. Il a été constaté que des stratégies de contrôle différentes pilote variant dans le temps semblent être distinguées par différents groupes, même si les groupes se croisent pendant une fenêtre temporelle donnée. Il a été appris que s'il existe une méthode disponible dans GMM pour déterminer le nombre optimal de grappes, le jugement des spécialistes pour proposer un nombre significatif et interprétable de grappes est plus important que ce que suggère BIC. Il a également été exploré que le regroupement basé sur les événements peut atteindre l'identification des valeurs aberrantes sans utiliser une «méthode de densité de probabilité de log» de GMM.

En résumé, l'approche GMM offre un moyen prometteur de regrouper les sessions de formation des pilotes et de détecter les vols anormaux. Le modèle peut prédire le modèle de comportement des stagiaires pilotes, fournissant l'algorithme à compléter par les formateurs pilotes qui examinent et attribuent une signification opérationnelle aux grappes. La méthode peut fournir des rétroactions à la volée aux formateurs pilotes grâce à une approche de formation basée sur les données, fondée sur des preuves et adaptative.

Mots-clés : Regroupement, Anomalie, Apprentissage non supervisé, Modèle de mélange gaussien, Évaluation automatisée des performances, Formation pilote basée sur les compétences

Abstract

This project applied a probabilistic class of artificial intelligence algorithms in an unsupervised learning domain called Gaussian Mixture Model (GMM) for clustering pilot training data and detecting flights with unusual data patterns. The approach can promote the capability of data-driven adaptive pilot training tools and to help the instructors select the most appropriate training exercises, ensuring targeted training recommendations.

To reach the research goal, a pipeline from dataset to GMM input was designed. Then, flight comparison models were studied, and one model was selected. For outlier detection, a "probability averaging method" was proposed and experimented against "log-probability density method" and the latter was selected. GMM performed on the processed data and results showed that GMM can identify clusters and detect anomaly among flights at both maneuver and event levels. It was found that different time-varying pilot control strategies appear to be distinguished by different clusters even if the clusters cross during a given time window. It was learned that while there is an available method in GMM to determine the optimal number of clusters, the specialists' judgment to propose a meaningful and interpretable number of clusters is more important than what GMM suggests. It was also explored that the event-based clustering can reach to outliers' identification without using a "log probability density method" from GMM.

In summary, the GMM approach provides a promising way to cluster pilot training sessions and to detect abnormal flights. The model can predict the pilot trainees' pattern of behaviour, providing the algorithm to be complemented by the pilot trainers who review and assign operational meaning to the clusters. The method can provide on fly feedbacks to the pilot trainers through a training approach that is data driven, evidence-based and adaptive.

Keywords : Clustering, Anomaly, Unsupervised Learning, Gaussian Mixture Model, Automated Performance Evaluation, Competency-based Pilot Training

Contents

List of Tables

List of Figures

List of Abbreviations

AIC : Akaike’s Information Criterion

AWS : Amazon Web Services

BIC : Bayesian Information Criterion

CBT : Competency-Based Training

DTW : Dynamic Time Warping

EBT : Evidence-Based Training

ED : Exceedance Detection

EM : Expectation Maximisation

FOQA : Flight Operational Quality Assurance

GMM : Gaussian Mixture Model

OEI : One Engine Inoperative

PDF : Probability Density Function

This research is dedicated to talented and generous Machine Learning Researchers who openly share their creativity to the public and allow others to learn.

Acknowledgments

I wish to express my sincere appreciation to Sandeep Subramanian, PhD candidate and my Mila supervisor, as well as Dr. Mikhail Klassen, my supervisor at Paladin AI, for their continues support and guidance throughout this research. Thank you to Daniel Spira, Jose Miguel, Aaron J. Maxwell and other colleagues at Paladin AI for sharing their knowledge and support. I would like to thank the Government of Canada, the Government of Quebec, and Paladin AI for granting me financial support. I would also like to thank Mitacs, a not-for-profit organization, that coordinated and facilitated this financial support. My most sincere appreciation is to the generous and talented individuals in internet open source online communities such as Stack Overflow who openly share their creativity to the public and allow others such as me to learn.

I wish to acknowledge and appreciate the support and great love of my family, my wife, Maryam; my son, Sam; and my daughter, Sara.

Chapter 1 – Introduction

In a trend towards increasingly complex aircraft, pilots need to be trained to obtain high level of competencies in situational awareness, problem solving, leadership and communication skills. Traditionally, pilot training methods includes classroom instruction, practical training and computer-based training. The training is performed in either the aircraft or a flight-training device or both. Over the last 30 years, the availability of flight operations and training activity data has increased significantly (IATA, 2013). Flight simulators in the training centres can use Flight Data Recorder (FDR) to record flight parameters such as altitude, airspeed, pitch etc onboard during the pilot training. This ability beside recent advancements and capability to use cloud-based machine learning algorithms as well as ability to use big data, have made pilot training possible using a data-driven training system. This system can leverage big data analytics to make pilot training more efficient and effective. Paladin AI, a pioneer in this field, has created a data-driven adaptive training tool which powered by machine learning. This tool, called "The InstructIQ Dashboard", detects pilot behavior and competency and can objectively assess airline pilot performance using real-time feedback during training sessions. Objective of the current research is to promote the capability of the data-driven adaptive pilot training tool and to make targeted training recommendations by using a clustering machine learning algorithm.

More specifically, the focus is on working with real time-series of flight parameters such as airspeed, pitch etc from a flight simulator for takeoff and landing activities. The data are extracted and used to develop a new analytics technique for clustering pilots' behaviour and to detect pilots' competency against performance norms and pilots' anomaly behavior. The utilized analytical technique is an unsupervised clustering method, called Gaussian Mixture Model (GMM). The GMM is applied on takeoff and landing activities to identify multiple markers of either good or poor pilot performance. It is used for clustering to detect the norm of flight operations and to identify the abnormal flights. The norm is characterized by clusters and the temporal distribution of these clusters. The project delivers a usable proof-of-concept that will be ultimately be deployed to a production environment by Paladin AI devops team.

Chapter 2 – Literature Review

The following sections of this chapter present literature reviews on clustering algorithms, flight's anomaly detection methods, and pilot training paradigms.

Clustering

Clustering include various unsupervised classification techniques to identify homogeneous groups in a data set among the observations. In clustering, we look for patterns in the data set in order to place them into separate groups in a way that observed data within each group are similar to each other and different than observed data in other groups (James, Witten, Hastie, & Tibshirani, 2017). There are different ways to perform clustering analysis, such as partition-based, hierarchy based, proximity-based, and so forth.

In this section, different types of clustering methods with one typical algorithm from each are described. The clustering methods are ended with GMM, which was used in this project.

K-means partitioning algorithms

The k-means algorithm (Hartigan & Wong, 1979), is the most popular clustering tool used in scientific and industrial applications. K-means is a point-assignment approach to partition a data set into K distinct and non-overlapping clusters. In K-means clustering, there are two assumption: (1) The observed data are partitioned into K pre-specified number of clusters. (2) The algorithm is assumed point assignment in a Euclidean space. The K-mean clustering follows the below algorithm (James, Witten, Hastie, & Tibshirani, 2017).

(1) A number, from 1 to K, is randomly assigned to each of the observations. These serve as initial cluster assignments for the observations.
(2) The below iteration is performed until the cluster assignments stop changing:

(a) For each of the K clusters, the cluster centroid is computed. The k^{th} cluster centroid is the vector of the p feature means for the observations in the K^{th} cluster.

(b) Each observation is assigned to the cluster with closest centroid based on the Euclidean distance definition.

Although the algorithm is guaranteed to converge, it converges to a local optimum and it depends on the centroid initialization. K-Means is one of the fastest clustering algorithms (Leskovec, Rajaraman, & Ullman, 2014).

Hierarchical algorithms

As opposed to in K-means clustering, which looks for partitioning the observations into a pre-specified number of clusters, in the hierarchical clustering, the number of clusters are not known in advance. Hierarchical clustering has a tree-based representation of the observations, called a dendrogram. There are two types of hierarchical clustering, Divisive (top-down) and Agglomerative (bottom-up and more common type), illustrated in Figure 1.

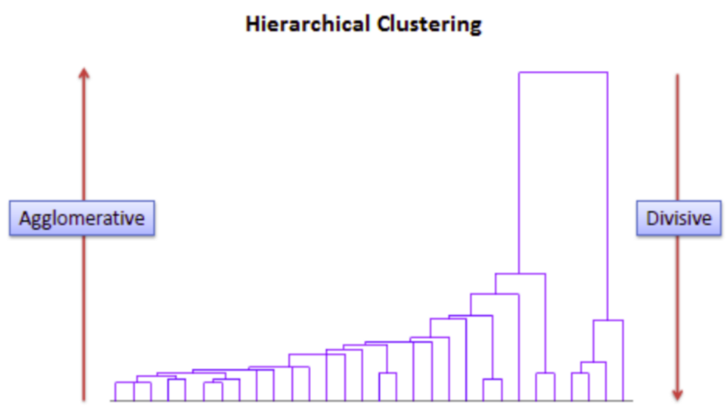


Figure 1 Hierarchical Clustering diagram, courtesy of saedsayad.com

In the agglomerative model, dendrogram (the upside-down tree) starts from the leaves and combining clusters up to the trunk. Each object is in its own cluster at the bottom of the hierarchy. Pairs of clusters are merged based on their similarity as one moves up the hierarchy, until all the objects are in a single cluster or until a specific termination conditions are met. The divisive approach works in the opposite way. The agglomerative algorithm is described below (James, Witten, Hastie, & Tibshirani, 2017).

(1) Using a measure such as Euclidean distance, pairwise dissimilarities among n observation (combination of 2 from n = n(n − 1)/2) is measured. Each observation is treated as its own cluster.

(2) For i = n, n − 1, . . . , 2:

(a) All pairwise inter-cluster dissimilarities among the i clusters are examined and the pair of clusters that are most similar are identified. These two clusters are combined and the dissimilarity between the two clusters is an indication for the dendrogram's height where combination is placed.

(b) The new pairwise inter-cluster dissimilarities are computed among the $i - 1$ remaining clusters.

Note that in the hierarchical clustering, user can choose in various ways which two clusters to merge next and decide when to stop the merging process.

The algorithm for the hierarchical clustering is not very efficient for big number of observations because its time complexity is $O(n^3)$.

Density-Based / proximity-based algorithms

DBSCAN (density-based spatial clustering of applications with noise), (Ester, Kriegel, Sander, & Xu, 1996), is a proximity-based algorithm that identifies clusters based on a density criterion. The density-based approaches find clusters by measuring how close a point is to its neighbours. It assumes that points in a cluster are close to each other and have similar proximity to its neighbours. The distance function needs to be defined for measuring the proximity. A cluster is formed if at least the minimum number of points are located within ε radius of a circle as shown in Figure 2.

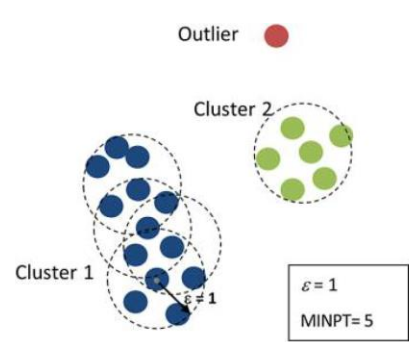


Figure 2 DBSCAN clustering process illustration

A formed cluster can grow if the added points in the neighborhood meet the same density criterion until no more point can be added. New clusters are formed if it satisfies the density criterion. The points that do not belong to any cluster are labelled by the DBSCAN algorithm as outliers. The method does not need a prior knowledge from the number of clusters in the data. Clusters are automatically formed until all data points have been processed.

DBSCAN has cited the most in scientific literature and is one of the most common clustering algorithms (From Wikipedia, 2020) . An abstract form of the algorithm is described below.

(1) The points in the ε neighborhood of every point are found, and the core points with more than min Pts[1] neighbors is identified.
(2) The connected components of core points on the neighbor graph is found while all non-core points are ignored.
(3) Each non-core point is assigned to a nearby cluster if the cluster is in ε neighbor, otherwise it is considered as noise.

CURE algorithm

CURE algorithm (Clustering Using REpresentatives) is an efficient data clustering algorithm for large databases. It assumes a Euclidean space. It does not have an assumption about the shape of clusters, i.e. there is no necessity to have a normal distribution. It can have any shape like strange bends, S-shapes, or even rings. Therefore, it is a robust model to outliers and able to identify clusters that have size variances and non-spherical shapes (Guha, Rastogi, & Shim, 2001). As the name of the algorithm implies, it uses a collection of representative points instead of representing clusters by their centroid.

The algorithm is described below (Leskovec, Rajaraman, & Ullman, 2014).

(1) A small sample of the data and cluster is taken in main memory. It is recommended to use a hierarchical method in which clusters are fused when they have a close pair of points.
(2) A small set of points from each cluster is selected as representative points. These points need to be as far as possible from each other.
(3) A fixed fraction of the distance between each of the representative points location and the centroid of its cluster should be moved. This step requires a Euclidean space.

[1] Min Pts is the minimum number of points required to form a dense region

(4) Two clusters if they have a pair of representative points need to be merged, one from each cluster, that are close enough. The merging step should be repeat, until there are no more sufficiently close clusters.

The algorithm time complexity is $O(n^2 \log n)$, and space complexity is $O(n)$.

Clustering in non-Euclidean spaces

GRGPF is an algorithm that does not require a Euclidean space. GRGPF refers to the authors name (Ganti, Ramakrishnan, Gehrke, Powell, & French, 1999). GRGPF uses both hierarchical and point-assignment approaches. The same as CURE algorithm, it represents clusters by sample points and the same as hierarchical algorithm it organizes the clusters in a tree structure like a B-tree. Therefore, a new data is located to the appropriate cluster when it passed down on the tree. Summaries of some clusters are held by the leaves of the tree, and the tree's interior nodes include subgroups of the information that describes the clusters that are reachable through that node. In this algorithm, the clusters are grouped based on their distance from each other, i.e. the clusters reachable from one interior node are close and also the clusters at a leaf are close. For more details, refer to (Ganti, Ramakrishnan, Gehrke, Powell, & French, 1999).

Gaussian Mixtures probabilistic clustering

In the non-probabilistic ("hard") clustering algorithms, each observed data belongs to a cluster and does not have its place in other clusters (a binary character). However, in the GMM algorithm as a probabilistic ("soft") algorithm, any observed data belongs to all clusters with different level of probabilities. Soft algorithms capture richer information from the data pattern than hard algorithms.

GMM is a parametric probability density function. It is represented as a weighted sum of Gaussian component densities where each component is a multivariate Gaussian that signifies a type of normal data (Reynolds, 2008). A data is considered as anomalies if it does not belong to any Gaussian component. The advantage of GMM compared to K-means is that it can provide statistical inferences for clusters, underlying distributions. Since GMM has flexibility and statistical features, it has been frequently used to model complex multivariate data for different applications, particularly in speech recognition (Reynolds, 2008).

In the following, the theory of GMM is explained (Reynolds, 2008), (Li & Hansman, 2013), (Li L. , Hansman, Palacios, & Welsch, 2016).

A GMM with K components (number of clusters) represents the probability density function (PDF) of a random variable, $x \in R^d$, as a weighted sum of k Gaussian distributions:

$$p(\mathbf{x}|\lambda) = \sum_{i=1}^{K} \omega_i g(\mathbf{x}|\mu_i, \Sigma_i)$$

where λ is the mixture model, ω_i corresponds to the weight of component i and the density of each component is given by the normal probability distribution:

$$g(\mathbf{x}|\mu_i, \Sigma_i) = \frac{1}{\sqrt{(2\pi)^M |\Sigma_i|}} e^{-\frac{1}{2}(\mathbf{x}-\mu_i)' \Sigma_i^{-1} (\mathbf{x}-\mu_i)}$$

During training, the parameters ω, μ and Σ are optimised iteratively through the Expectation Maximisation (EM) algorithm (Dempster, Laird, & Rubin, 1977) in order to maximise the log-likelihood of the model. Given a group of n independent and identically distributed samples $X = \{x_1, x_2, ..., x_n\}$, the log-likelihood corresponding to a mixture model λ is given by

$$\ln(p(\mathbf{x}|\lambda)) = \sum_{t=1}^{N} \ln\left(\sum_{i=1}^{K} \omega_i g(\mathbf{x}_t|\mu_i, \Sigma_i)\right)$$

The process starts with an initial model ($\lambda = \{\omega_i , \mu_i, \Sigma_i\}$, i = 1, …, K). The goal is to estimate a new model which has a larger likelihood for the given training data. After that, the new model is considered as the initial model for the next iteration. The process is repeated until some convergence threshold is reached.

Selection of Optimum Number of Clusters

To use GMM, number of clusters needs to be specified in advance, i.e. you know the distribution of your model and can guess number of Gaussian distributions. If that number can not be guessed, then optimal number of clusters can be determined by a sensitivity analysis method such as

Akaike's Information Criterion (AIC) and Bayesian Information Criterion (BIC) (Schwarz, 1978). Among several sensitivity analysis methods to define the optimum number of clusters in GMM, BIC outperforms (Steele & Raftery, 2009). AIC and BIC accompanying by libraries for GMM clustering are available tools in Scikit-learn.org.

BIC is a measure of the relative goodness of fit of a statistical model. BIC tests several values of K and selects the minimum value of K that attains adequate accuracy as the optimum number of clusters. The lower is the BIC, the better is the model. When number of K is increased the likelihood is increased but doing so will result in overfitting. In order to avoid overfitting, BIC penalizes models with a big number of clusters. BIC is used in this research to determine optimum number of clusters, K, for running GMM.

Selection of Covariance Type

GMM default covariance types in Scikit-learn.org is "full". There are four options for types of covariance: (1) "full", where each component has its own general covariance matrix; (2) "tied", where all components share the same general covariance matrix; (3) "diag", where each component has its own diagonal covariance matrix; and (4) "spherical", where each component has its own single variance. In this research, we obtained more meaningful results using diagonal covariance matrices, independent parameters assumption among Gaussian components. Diagonal covariance matrices reduce computational complexity. The component Gaussians are acting together to model the overall vector. For this reason, full covariance matrices are not required even when the flight parameters are not statistically independent (Li L. , Hansman, Palacios, & Welsch, 2016).

Flight's Anomaly Detection

Definition of the anomaly flights emerges from the normal flights. Normal flights belong to mainstream of identified clusters. Every cluster is representative of a frequently observed operational mode in the dataset. We can refer to it as a nominal mode. The identified flights' clusters can be used for the assessment of the abnormal flights. Many technical parameters such as airspeed, altitude, pitch, roll, engine parameters, etc. are recorded throughout a flight or flight simulator by Flight Data Recorder (FDR). The recorded dataset includes rich information about pilot operations. The current industry standard methods for detecting anomalies in flight data are Exceedance Detection (ED), and three other data-driven methods, MKAD, ClusterAD-Flight and ClusterAD-DataSample (Li L. , Hansman, Palacios, & Welsch, 2016). ClusterAD-DataSample, which is a GMM anomaly detection method for flight operation and safety monitoring, used in this research for pilot training. In the following, above mentioned four methods are described.

Exceedance Detection (ED)

ED is a standard method widely used by airlines in Flight Operational Quality Assurance (FOQA[2]) programs. In ED, pre-defined operationally undesired events are detected. Acceptable limits of list of flight parameters such as the pitch at takeoff, the speed at takeoff climb, the time of flap retraction etc. are predetermined by safety specialists and the parameters are monitored to check if the flight parameters exceed the predefined limits. If the limits are exceeded, the event is labeled in three alarm levels based on the severity of the deviation from expected values (Li L. , Hansman, Palacios, & Welsch, 2016).

Multiple Kernel Anomaly Detection (MKAD)

MKAD is a method developed by (Das, Matthews, Srivastava, & Oza, 2010) based on one-class Support Vector Machine (SVM). The method combines both continuous and discrete parameters via kernel functions. It can combine information of different data types and detect anomalies by analyzing various parameters simultaneously. MKAD assumes exitance of only one single consistent data pattern for normal operations, which does not hold in real practice.

ClusterAD-Flight

ClusterAD-Flight is a method invented by (Li, Das, Hansman, Palacios, & Srivastava, 2015). For cluster analysis, this method uses DBSCAN algorithm, a non-parametric clustering techniques explained on page 16. It can detect abnormal flights during phases of flight that have standard procedures associated and clear time anchors such as take-off or landing approach, rather than

[2] FOQA is a method of capturing, analyzing and/or visualizing the flight's data.

instantaneous abnormal data samples during a flight. The method can automatically determine multiple clusters and detect outliers.

ClusterAD-DataSample

ClusterAD- DataSample is a method invented by (Li L. , Hansman, Palacios, & Welsch, 2016). For cluster analysis, this method uses GMM based clustering, a probability density function cluster analysis technique explained on page 18. GMM used on digital flight data in order to automatically recognize multiple typical patterns of flight operations and to detect flights with uncommon data patterns. ClusterAD-DataSample can identify clusters from data samples at each time point during a flight.

Pilot Training Paradigms

Pilot training consists of basic training and type-rating. Training methods includes classroom instruction, computer-based training and practical training. These methods of training are performed either in the aircraft or in a flight-training simulator. Since mandatory recurrent training cycles with high cost is a necessity in the aviation industry, the industry has become an early user of the computer-based training technologies such as the flight simulators, which has lower cost than practical training. The chosen pilot training method, effects on the aviation safety as well as Airlines cost (Klassen, 2020). In the following, pilot training approaches are explained.

Competency-based approach

Competency-Based Training (CBT) approach is characterized by a performance orientation. In the training and assessment development, the emphasis is on standards of performance. (IATA, 2013). Competency-based training programmes and harmonized task performance standards result in personnel who are trained and assessed against competency standards. These types of trainings generate records showing the trained pilots meet the defined aviation standards and it makes easier for organizations to employ competent pilots. (ICAO, 2014).

Evidence-based training

Evidence-based training (EBT) appeared from the need to develop a new approach for competency-based training and assessment of airline pilots which is based on evidence. Rather than competency-based assessment which measures the pilot performance on individual events or maneuvers, the training and assessment in EBT is characterized by developing and assessing the overall capability of a trainee across a range of competencies. The EBT objective is to identify, develop and evaluate the essential competencies that pilots need in order to operate safely, effectively and efficiently. It is managed by collecting the most relevant threats and errors' evidences in operations and training. Over the last twenty years, the availability of data covering both flight operations and training activity has improved substantially. The significant improvement and availability of data in flight operations and training activity has proved and facilitate the need for the EBT approach (IATA, 2013).

Adaptive training

Since individual pilots has different experiences and background and learns at a different pace, a "one-size-fits-all" approach, practiced traditionally, is no longer beneficial. Meanwhile, technology facilitate the pilot training toward an adaptive training direction and to go for more personalized approaches. Adaptive approach is an optimised way to customize training for each pilot. Since pilot trainees have individual difference and they respond differently to challenging circumstances, machine learning algorithms can be trained to detect and measure pilots' competencies and recognize their differences (Klassen, 2020).

InstructIQ, an evidence-based and adaptive tool

Paladin AI created an evidence-based and adaptive tool, called "InstructIQ dashboard"[3]. This dashboard presents flight trainers with a Training summary radar chart which includes eight pilot competency components, shown on Figure 3.

- Procedures
- Flight Path (Manual)
- Flight Path (Automatic)
- Workload
- Awareness
- Problem Solving
- Leadership
- Communication

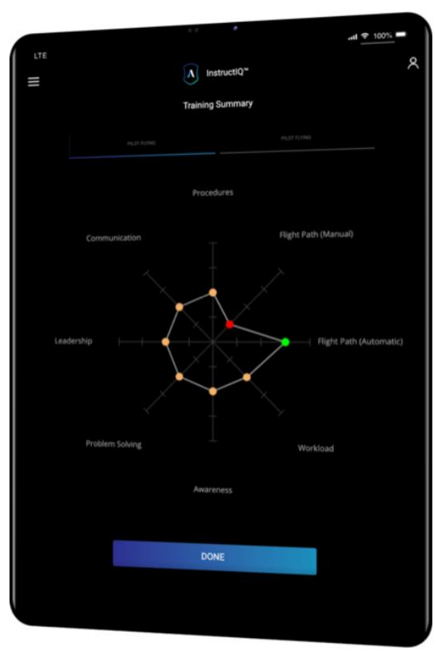


Figure 3 Paladin AI InstructIQ dashboard

Theses components are based on IATA evidence-based training implementation guideline published on (IATA, 2013) and is a solution for adaptive aviation training. The IATA guideline

[3] https://paladin.ai/product

associats multiple behavioral indicators with each competency component (listed on Appendix A of the IATA report). Pilot instructors can use InstructIQ to monitors the pilot's competency component in real time and to receive the training analytics and competency feedback (Klassen, 2020). This research uses GMM as an analytics technique for inferring pilot competency that can be utilized within InstructIQ.

Chapter 3 – Methodology

Investigated Flight's activities

Takeoff

Takeoff in this research is defined the flight's maneuver bounded between two events: liftoff and flap retraction. Liftoff happens when the aircraft wheels leave the ground. Flap retraction happens when the flap lever position at the end of takeoff activity is changed to zero. Duration of takeoff activity for different flights are different and in a range between 30 to 475 time-steps[4].

Landing Flare

The landing flare in this research is defined the flight's final approach maneuver which starts from 50 ft altitude and ends at the ground touchdown. Duration of landing flare activity for different flights are also different.

Data

Data Set

The landing and takeoff models were trained with Citation-XLS[5] pilot training data. The following features were extracted and used for the GMM trainings of takeoff and landing activities.

Continues variable

cas (computed airspeed, an absolute airspeed)

v2 (v2 is the speed at which the aircraft may safely be climbed with one engine inoperative.)

cas_minus_v2 (relative airspeed for takeoff: cas – v2)

CAS_Dtd (relative airspeed for landing flare: cas – speed at touchdown)

radalt (radar altitude, altitude relative to the beneath ground)

pitch

Discrete variable

flap_lever_pos (flap lever position)

[4] Every 3 time-steps are equivalent to one second.

[5] Citation-XLS is the Cessna Citation Excel, an American midsize business jet built by Cessna.

Data Extraction and Preprocessing

Access was granted to Paladin AI 'sandbox' development account on Amazon Web Services (AWS), using a Jupyter Notebook environment (Amazon Sagemaker) backed by multicore and high-memory compute intensive machines. For the python kernel on Sagemaker, conda_python3 was used. Pilot training data from Citation-XLS aircraft was available in AWS S3. In this research, the most available and ready to use ("labelled") data were used, i.e. 70 flights for takeoff activity and 91 flights for landing activity.

First a spark session was created. Then, the data was read into the spark dataframe. SQL commands were used to fetch the required flight parameters from the data. The data was converted to a pandas dataframe. Then, the following preprocessing were performed on the extracted data for the takeoff activity:

(1) Flights takeoff with v2 = 0 through the activity were identified as not correct recorded data and removed from the dataset.
(2) Flights session's starting time was considered as the primary key of the dataset to identify the unique flights.
(3) The variations of takeoffs activities were separated into two major categories: (1) full engine operative, (2) one-engine-inoperative (OEI). In Table 1, all takeoffs except "Single Engine Take-off" are considered as full engine operative.

(4) Table 1 Variations of takeoff

Name	Paladin AI Name	Description
Normal Take-off	TAKEOFF_NORMAL	Normal take-off
Single Engine Take-off	TAKEOFF_OEI	Take-off with a single engine running
Rejected Take-off	TAKEOFF_REJECTED	Rejected take-off
Instrument Take-off	TAKEOFF_INSTRUMENT	Take-off using instruments, usually in low visibiliy situations, visibility less than ~ 1 mile
Crosswind Take-off	TAKEOFF_CROSSWIND	Take-off with wind hitting the side of the airplane
Windshear Take-off	TAKEOFF_WITH_WINDSHEAR	Take-off with extreme wind patterns defined by FAA

(5) Fights time-steps at liftoff were identified

(6) Number of timesteps from liftoff to the end of takeoff for the shortest and longest flight were determined.

Note (1) for the takeoff activity, from 70 flights 21 are OEI flights and others are flights without an engine failure.

Note (2) for the landing activity, there are 91 flights.

Next section explains how the extracted and processed data are prepared to feed GMM for both event and activity.

GMM Input

GMM input for event

Events such as liftoff (when the aircraft wheels leave the ground) or flap retraction (when the flap lever position at the end of takeoff activity is changed to zero) happens at one timestamp. GMM input for an event is a 2d array of flight parameters values with (Flights X Parameters) dimension where the array rows are flights and the columns are flight's parameter. Below is a typical example of the GMM 2d array input with 3 flight parameters for n flights at one time-step.

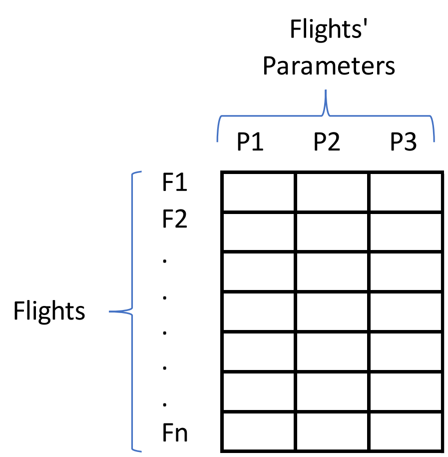


Figure 4 GMM input 2d array format for an event at one time-step

GMM input for activity

GMM input for activity is a 2d array of flight parameters values for entire manoeuvre (takeoff or landing activities). Dimension of an input 2d array for one parameter is (Flights X time-steps). For more than one parameter, time-steps are multiplied by the number of parameters. Below is a typical example of the GMM 2d array input with 3 flight parameters for n flights during m time-steps.

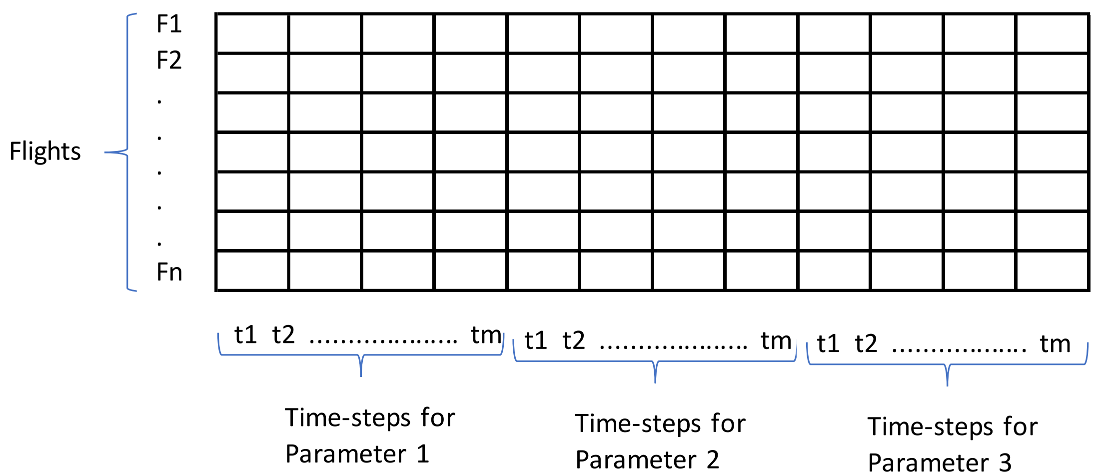


Figure 5 GMM input 2d array format for an activity during m time-steps

Sample Comparison Methods

Duration of an activity is different among different pilots. For example, the flight minimum duration of the takeoff activity has 30 time-steps (10 seconds) and the flight with maximum duration has 475 time-steps (158 seconds). Similar idea is applied for the landing activity. In this research, to make different flights with different durations comparable, two approaches were used: (1) same time-steps method (used for takeoff); (2) align time-steps method (used for landing).

Same Time-Steps Method

In this method, only flights' information at a specific number of time-steps are used. For analysis of the quantitative parameters (airspeed, pitch and altitude), the flights' parameters at the first 30 time-steps (time-steps of the flight with shortest takeoff) for all flights are compared. For analysis of the discrete parameter (flap lever position), the flights' parameters for 475 time-steps (time-steps of the flight with longest takeoff) for all flights are compared. Note that for the flap lever position, the takeoff activity restriction is relaxed and for some flights the activity is extended to climbing and cruise.

Aligned Time-Steps Method

In order to make flights' parameters with different number of time-steps comparable, the flights' time steps need to be aligned, i.e. the flights parameters vectors with different lengths to be adjusted and fitted to one size vector, i.e. a vector with a specific number of time-steps. In this research, two different approaches were experimented and compared: (1) Dynamic Time Warping (DTW); (2) Interpolation. Interpolation approach was selected and used for the landing flare activity. The explanation of two approaches are presented below and followed with a conclusion.

DTW

DTW is an algorithm to measure similarity between two temporal sequences, which may vary in speed. It calculates an optimal match between two given sequences using a minimum-distance warp path between two sequences (Salvador & Chan, 2004).

There is a python library available for DTW[6] that was used to align the individual flight vectors. Below is an alignment illustration between two typical flight sessions, session run id 16 with 111 timestamps and session run id 21 with 131 timestamps.

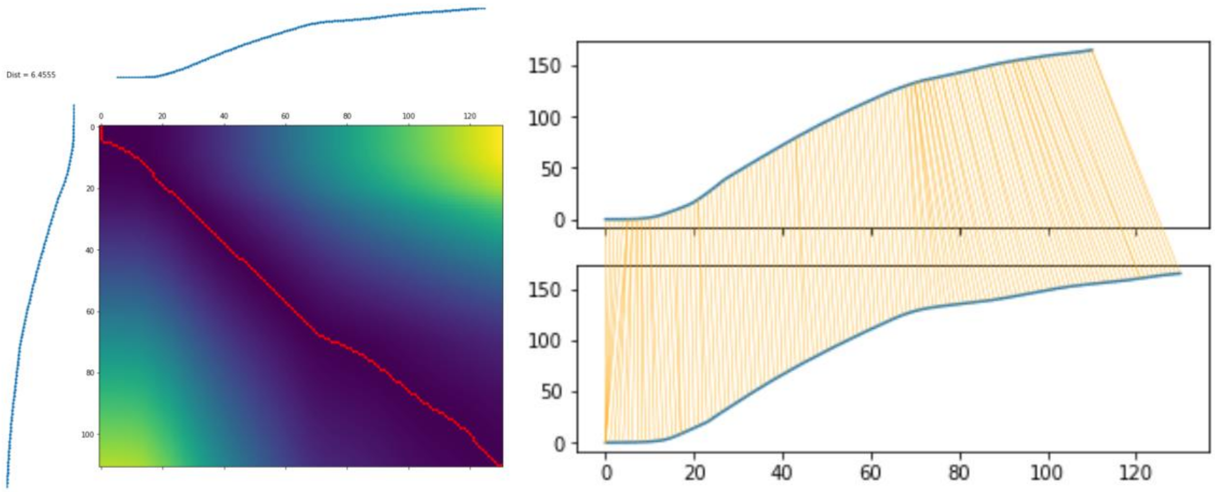


Figure 6 DTW alignment illustration for two flights

[6] from dtaidistance import dtw

In the following, CAS of 13 flights were experimented to align to a same number of time steps (142 time-steps belong to the flight session run id 36). Figure 7 illustrates the original and aligned CAS.

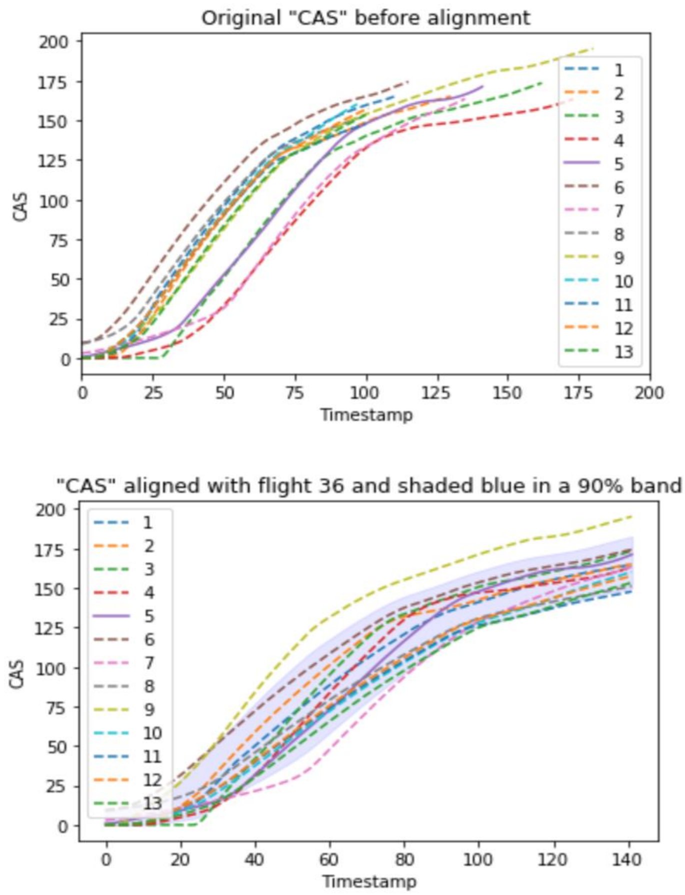


Figure 7 Flights' cas before and after alignment with DTW method

Figure 7 shows that the aligned CAS curves have the similar trend and curvatures as the original curves. It is noted that in the original "CAS" plot, flights 4 and 6 are located at the extremes, while in the aligned "CAS" plot (shaded in a 5%-95% band), flights 7 and 9 are out of the band. Note flight 9 has the longest running time.

Interpolation

Below is an illustration of CAS vs timesteps of a flight before and after the linear interpolation alignment. Before the alignment, there are 301 time-steps, liftoff happens at t=67 and CAS=123 After the alignment, there are 180 steps, liftoff happens at t=40 (67*180/301) and CAS=130

```
CAS Plot before and after alignment
Flight session run id = 1
Ativity code = TAKEOFF_WITH_WINDSHEAR
```

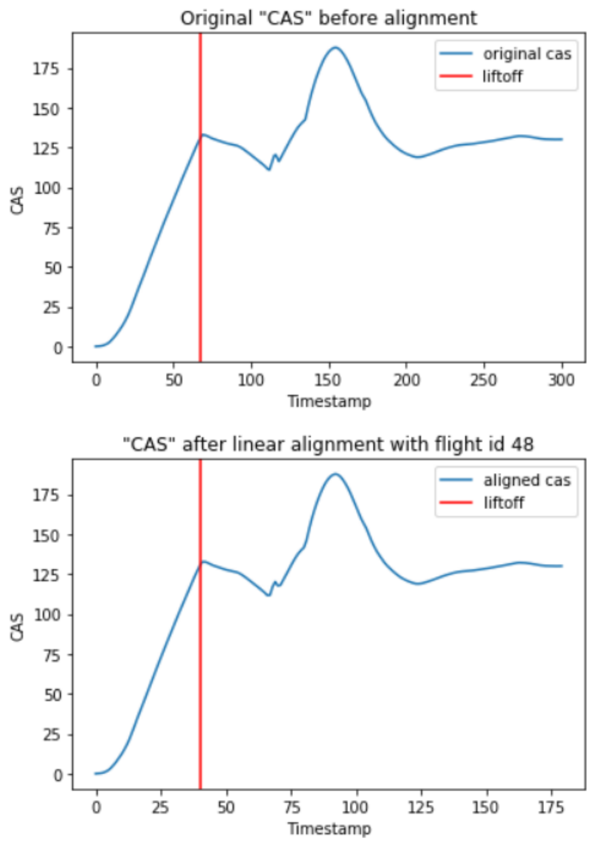


Figure 8 Flight's cas before and after alignment with Interpolation method

Conclusion

While DTW has a non-linear alignment advantage, it can be applied to only a pair of flights. To apply DTW to multiple flights, it is needed a reference flight to be chosen. It was noted that warping is sensitive on the chosen reference flight. Usage of DTW is costly as it has $O(n^2)$ complexity and it needs to be iterated over all the flights which are paired with the reference flight. Linear interpolation is more efficient than DTW method. At the same time, the trend of the CAS variations over the time are not much disturbed after a linear interpolation adjustment. Therefore, it was decided to use the linear interpolation as an adequate method to satisfy the required alignment. For the interpolation, a SciPy method was imported (*from scipy.interpolate import interp1d).*

Model

GMM (explained on page 18), was used in this research to determine flights' clusters and outliers.

Determine Optimum number of Clusters

In order to determine the optimum number of clusters, BIC method was used. For BIC explanation, refer to page 19. "GaussianMixture" and "bic" methods from Scikit Learn library were used on K different number of clusters from 1 to 10 and the K with minimum BIC was selected as the optimum number of clusters.

Identify Clusters

The determined optimum number of clusters, K, was used in the following simplified code to predict flights' clusters.

- from sklearn.mixture import GaussianMixture
- gmm = GaussianMixture(n_components=K)
- gmm.fit(param_groundoff)
- cluster = gmm.predict(param_groundoff)

Note "param_groundoff" is name of the input data. For input data explanation, refer to page 28.

Detect Outliers

In order to detect outliers, first a method referred to as "Probability Averaging Method" was planned and tried. Since the method did not reach to a meaningful result, another method referred to as "Log Probability Density Method" was used. Since the latter method reached to reliable results, for outlier flights detection in this research the "Log Probability Density Method" was used. In the following both methods are explained.

Probability Averaging Method

To identify the flight outliers by the planned probability average method, GMM was run **m**[7] times over **n** flights and probability values were determined. Then, for each flight and at each cluster

[7] M is number of time-steps during the activity.

the probability values were averaged. Dimension of the average probability matrix is: Flights X Clusters. Then at each cluster, a Gaussian shape was fitted to the average probability values of **n** flights. For each flight at each cluster, a z_score was calculated. A z_score threshold was defined and the flights' z_scores were compared with that threshold. Any flight that it's z_score was out of the threshold for all the clusters was labelled as an outlier flight.

In the following, the necessary steps to determine outlier flight are summarized. The example, used pitch as the input parameter.

- Fill the **n Flights X m time_steps X k Clusters** 3-d matrix with pitch probability values. At each time step, run a GMM to get k vectors of probability values for each flight. During the ground off, there are m time steps. Run GMM for each m time steps and fill the matrix with probability values.
- Calculate mean and variance for each row of flight and cluster. The output is a **n Flights X k Clusters** 2-d matrix filled with 2 values (P_ave and P_var) at each cell.
- Fit a gaussian shape to each cluster vector (find mean and variance parameter for each column).
- Use pitch values of a new flight and consider a threshold to determine if the flight is an outlier. If the flight is not an outlier, determine which cluster it does belong to and with what level of probability.

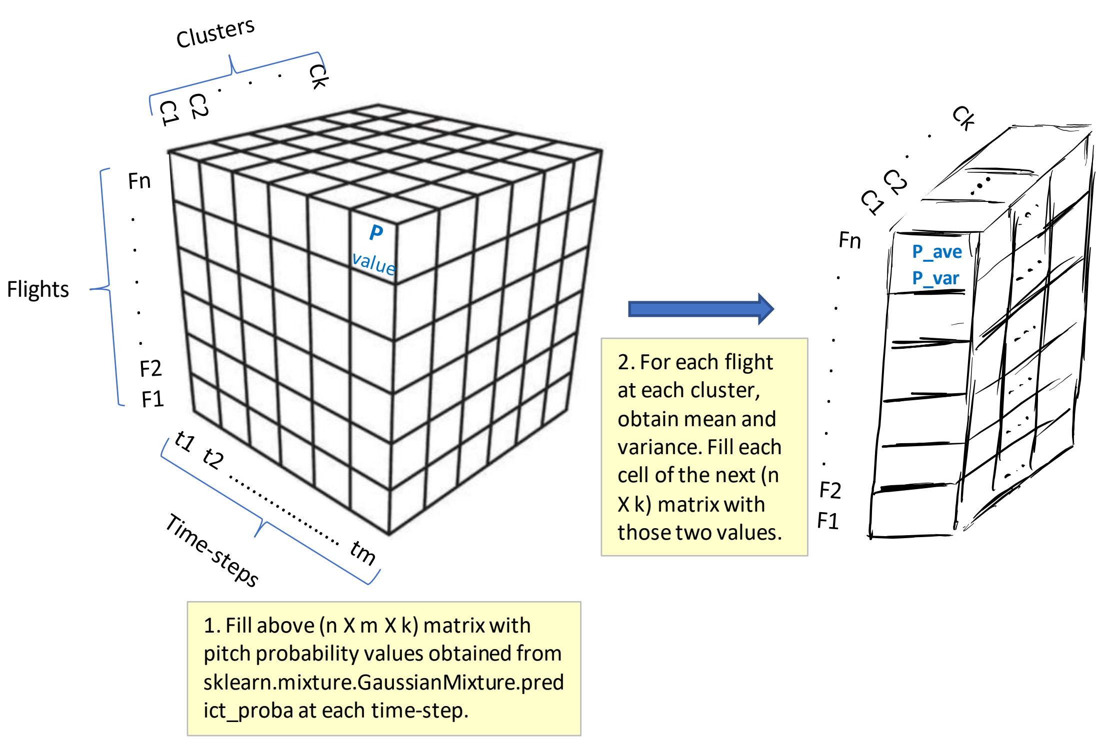
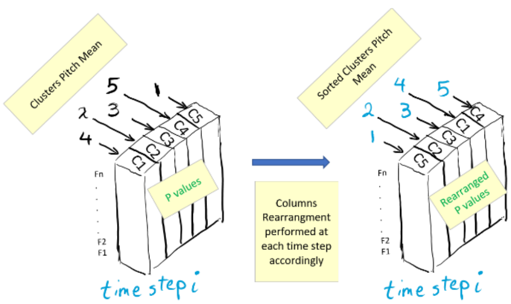


Figure 9 Probability Averaging Method illustration

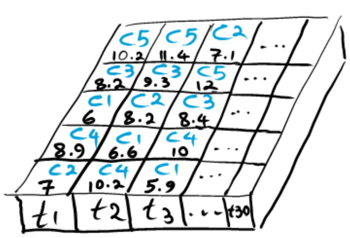
Note in the above model the clusters at each time-step rearranged as it is explained below.

It was explained that GMM performed on flights at each individual time step and the 3d matrix of (time-steps X flights X clusters) filled with the flight parameter's probability values. Normally, the clusters that are obtained at one time step are independent from clusters that are found at another time step. Therefore, there is no guarantee that the k clusters obtained for time step n have the same clusters' centres and also with the same order as the k clusters obtained for time step m. In the next step, for each flight and at each cluster we need to aggregate P_values. Therefore, we need a method to correct the order of the clusters at each time step. A proposed solution to alleviate the problem is reordering the columns at each time step so that the P_values follow an ascending order of the clusters pitch means, as shown below and exemplified for the first three time-steps considering pitch as the investigated flight parameter.



P values for the first 3 flights at the first 3 time steps:

Cluster centres
- GMM performed at each time step
- Clusters are named ascendentely



Probability results
- represents membership to each cluster

t1		c2	c4	c1	c3	c5
	F1	0.82	0	0.18	0	0
	F2	0.07	0	0	0.93	0
	F3	0.66	0	0.34	0	0
t2		c4	c1	c2	c3	c5
	F1	0	0.54	0.46	0	0
	F2	0	0	1	0	0
	F3	0	0.17	0.83	0	0
t3		c1	c4	c3	c5	c2
	F1	0	0	0.46	0	0.54
	F2	0	0	0.97	0	0.02
	F3	0	0	0.94	0	0.06

P-values are rearranged at each time step
They follow an ascending order of the pitch mean

t1		c1	c2	c3	c4	c5
	F1	0.18	0.82	0	0	0
	F2	0	0.07	0.93	0	0
	F3	0.34	0.66	0	0	0
t2		c1	c2	c3	c4	c5
	F1	0.54	0.46	0	0	0
	F2	0	1	0	0	0
	F3	0.17	0.83	0	0	0
t3		c1	c2	c3	c4	c5
	F1	0	0.54	0.46	0	0
	F2	0	0.02	0.97	0	0
	F3	0	0.06	0.94	0	0

Figure 10 Probability Averaging Method illustration - Clusters Rearrangement

Log Probability Density Method

To identify the outlier flights, the log of the probability density function (PDF) for any flight's parameter can be estimated using the **score_samples**() method from Scikit Learn library. The utilized code is shown below.

sklearn.mixture.gaussian_mixture.**score_samples**(param_groundoff)

Note "param_groundoff" is name of the input data. For input data explanation, refer to page 28.

Anomalies are flights with the least dense probabilities. An arbitrary density threshold of 10% was defined in this research. With this defined threshold, the flights located in the 10% low-density regions could be detected.

Chapter 4 – Results and Conclusion

In this chapter, GMM results for one and multiple parameters of flight are presented for takeoff and landing flare maneuvers. For the takeoff activity, sample size (number of flights) are 70. When quantitative parameter (airspeed, pitch and altitude) are used, the number of attributes (time-steps) is 30 for each parameter. When discrete parameter (flap lever position) is added to quantitative parameters, the number of attributes (time-steps) is 475. For the landing flare activity, sample size (number of flights) is 91 and number of attributes (time-steps) is 50.

In the following, Results and Discussion section is presented and followed by Conclusion and Future Works sections.

Result and Discussion

In this section, results for time-series clustering and event clustering are presented and discussed.

Takeoff Maneuver (30 time-steps)

In this section, results of GMM analysis performed on different parameters (flap lever position, altitude, airspeed), individually and together. Number of GMM attributes (time-steps) in this section is 30 (length of the shortest flight).

Airspeed

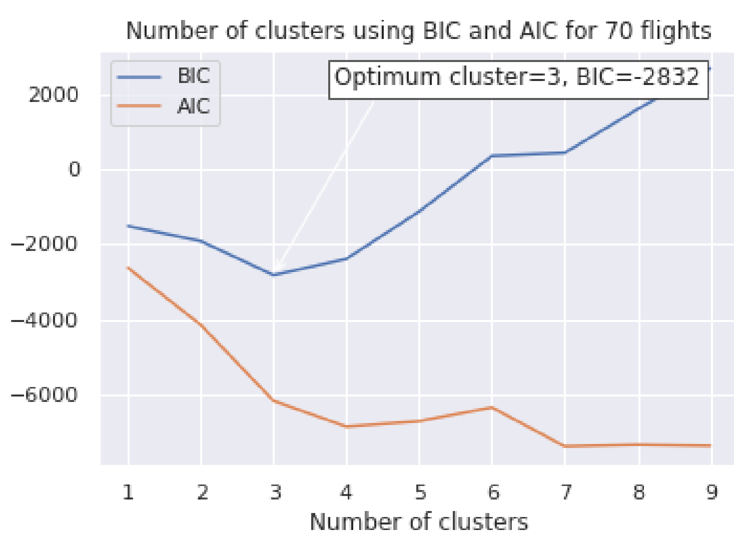


Figure 11 Airspeed (Flights Optimum Number of Cluster)

BIC determined 3 clusters as the best number of clusters. For BIC explanation, refer to page 19.

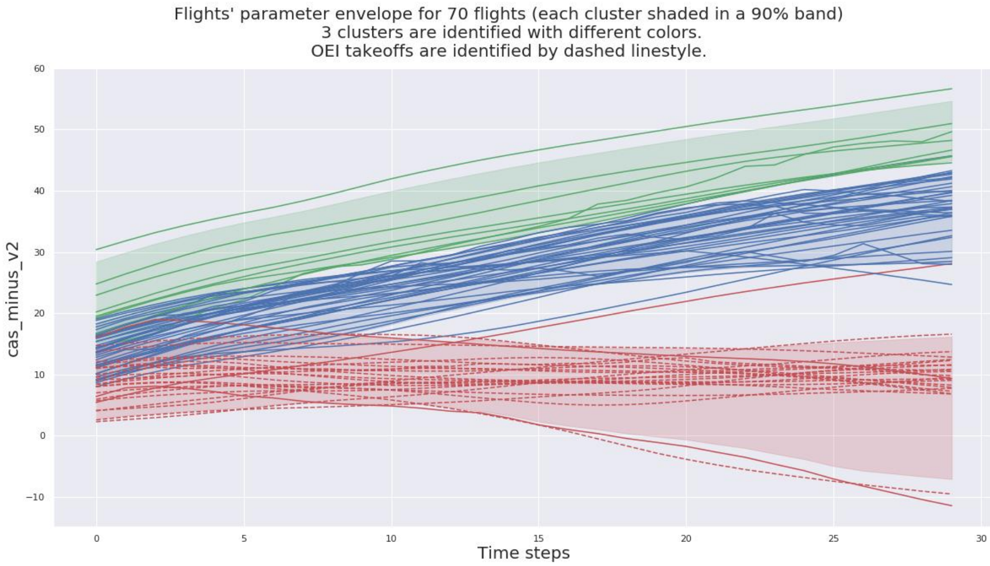


Figure 12 Airspeed (Flights Clusters)

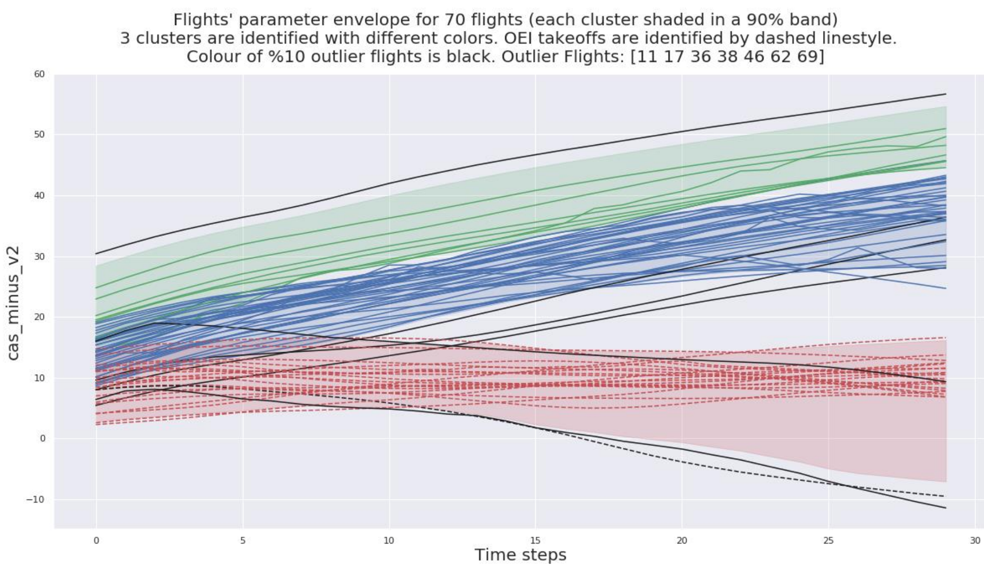


Figure 13 Airspeed (Flights Outliers)

GMM separated flights into three clusters. Note majority of the OEI flights (dash style lines) have been detected in red color cluster. Figure 18Figure 13 is the same as Figure 17Figure 12 with an additional identification of 10% anomaly flights with black color. It seems outliers are the flights either different than the trend of the majority flights inside the clusters or they are at the border of clusters.

Pitch

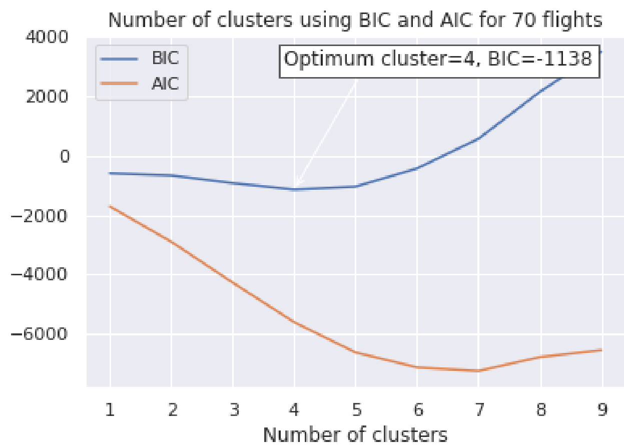


Figure 14 Pitch (Flights Optimum Number of Cluster)

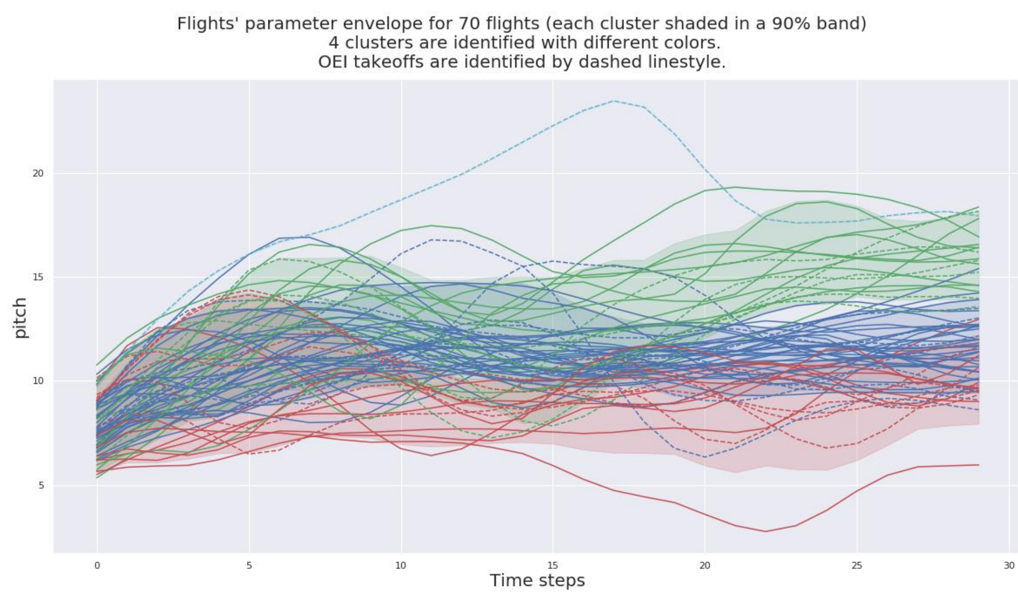


Figure 15 Pitch (Flights Clusters)

From Figure 15, it is observed that different time-varying pilot control strategies appear to be distinguished by different clusters even if the clusters cross during a given time window. The clusters seem to have the following meaning:

(1) Cyan: one cluster assigned to one flight with the highest pitch value. Note that in a separate experiment it was observed that when the GMM number of clusters changed from 4 to 3, the mentioned flight is located in a cluster similar to the green color cluster in the 4 clusters experiment.
(2) Green: Normal pitch at liftoff, migrating to high pitch starting from 5-10 time-steps
(3) Blue: Generally, 10-12 deg pitch. This is the normal performance cluster.
(4) Red: Starts with a medium to high pitch and after around 5 time-steps decreases and continues to low pitch.

Figure 18Figure 15 is the same as Figure 15 with an additional identification of 10% anomaly flights with black color.

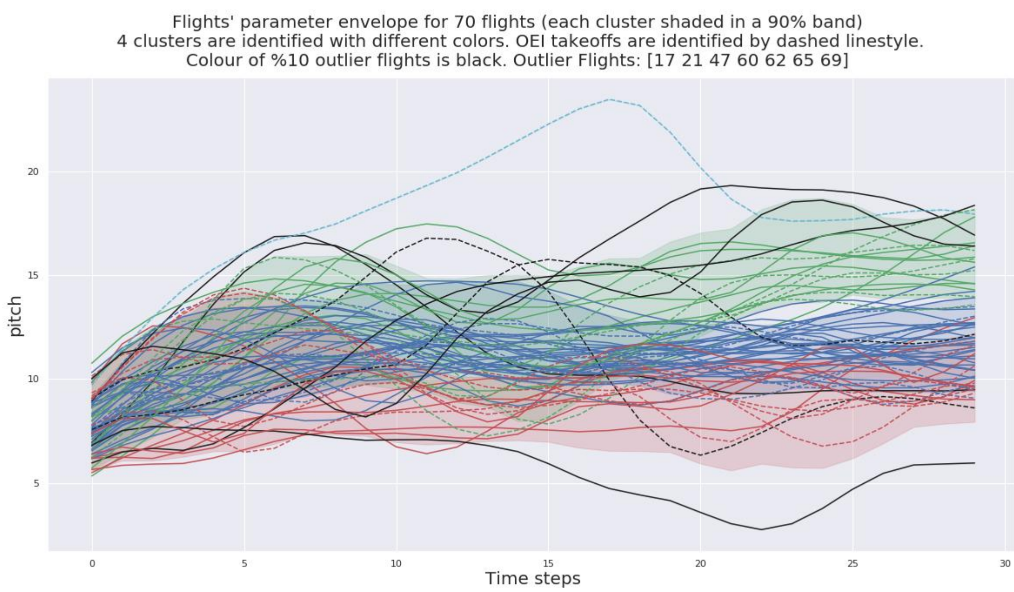


Figure 16 Pitch (Flights Outliers)

Altitude

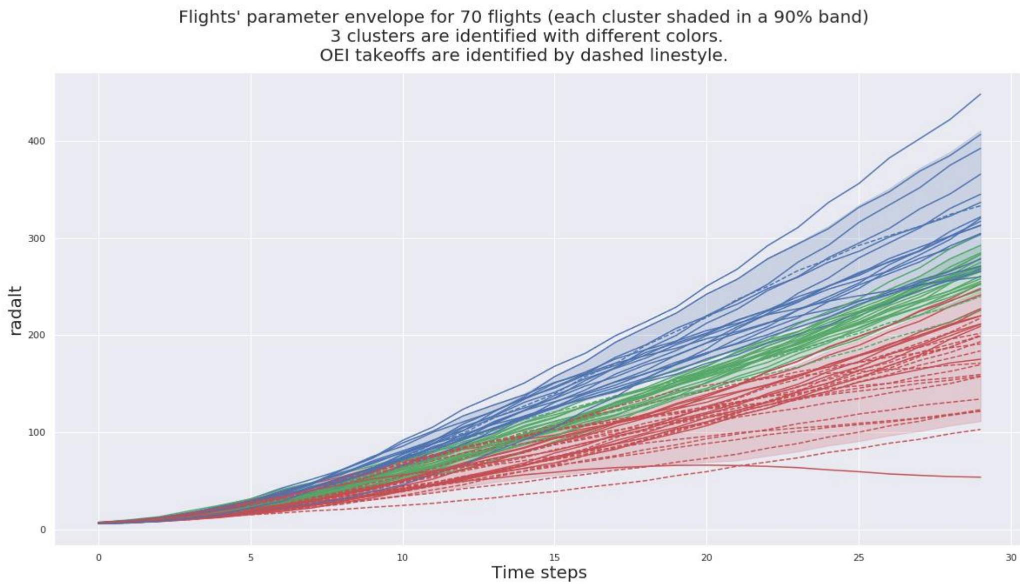


Figure 17 Altitude (Flights Clusters)

3 clusters were assigned for the GMM analysis of altitude. Flights are classified to high, medium and low altitudes. Majority of the OEI flights (dash style lines) have been detected in the low altitude cluster (red color lines). Note that in a separate experiment it was observed that when the GMM number of clusters changed from 3 to 2, the flights in the green cluster were separated into red and blue clusters.

Figure 18 is the same as Figure 17 with an additional identification of 10% anomaly flights with black color.

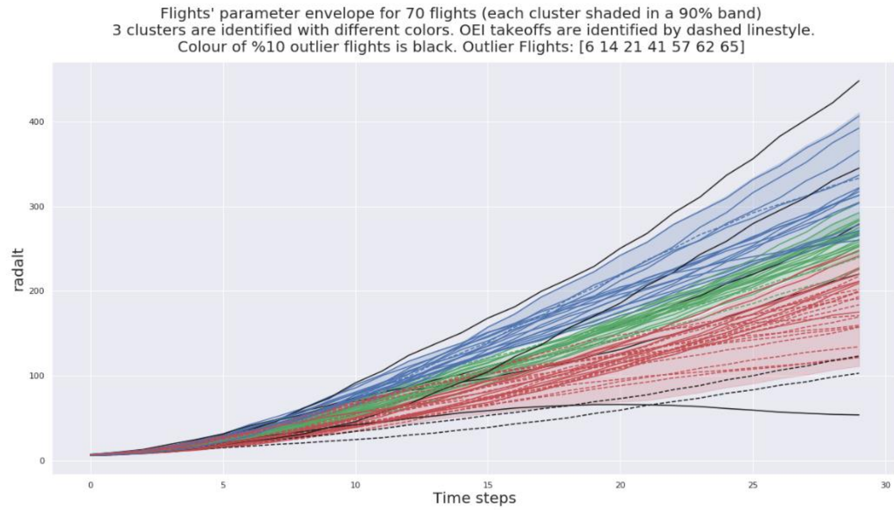


Figure 18 Altitude (Flights Outliers)

Airspeed and Altitude

The following plots and discussion are based on GMM analysis where two parameters (airspeed and altitude) have been used, i.e. number of features are 30 time-steps X 2 parameters = 60. Therefore, GMM input data is 70 flights X 60 attributes. Results are shown for 30 time-steps.

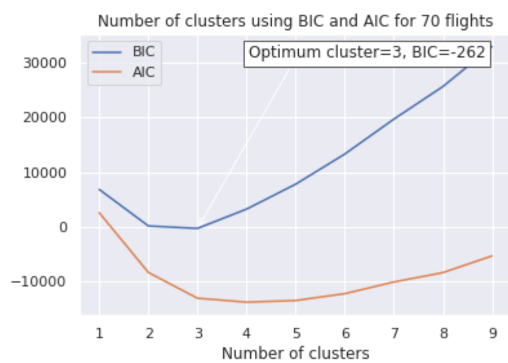


Figure 19 Airspeed and Altitude (Flights Optimum Number of Cluster)

Since Figure 19 shows a minor difference between BIC values for 2 and 3 clusters, GMM analysis is performed for 2 clusters. Figure 20 demonstrates flights airspeed-altitude scatter plots at each time-steps.

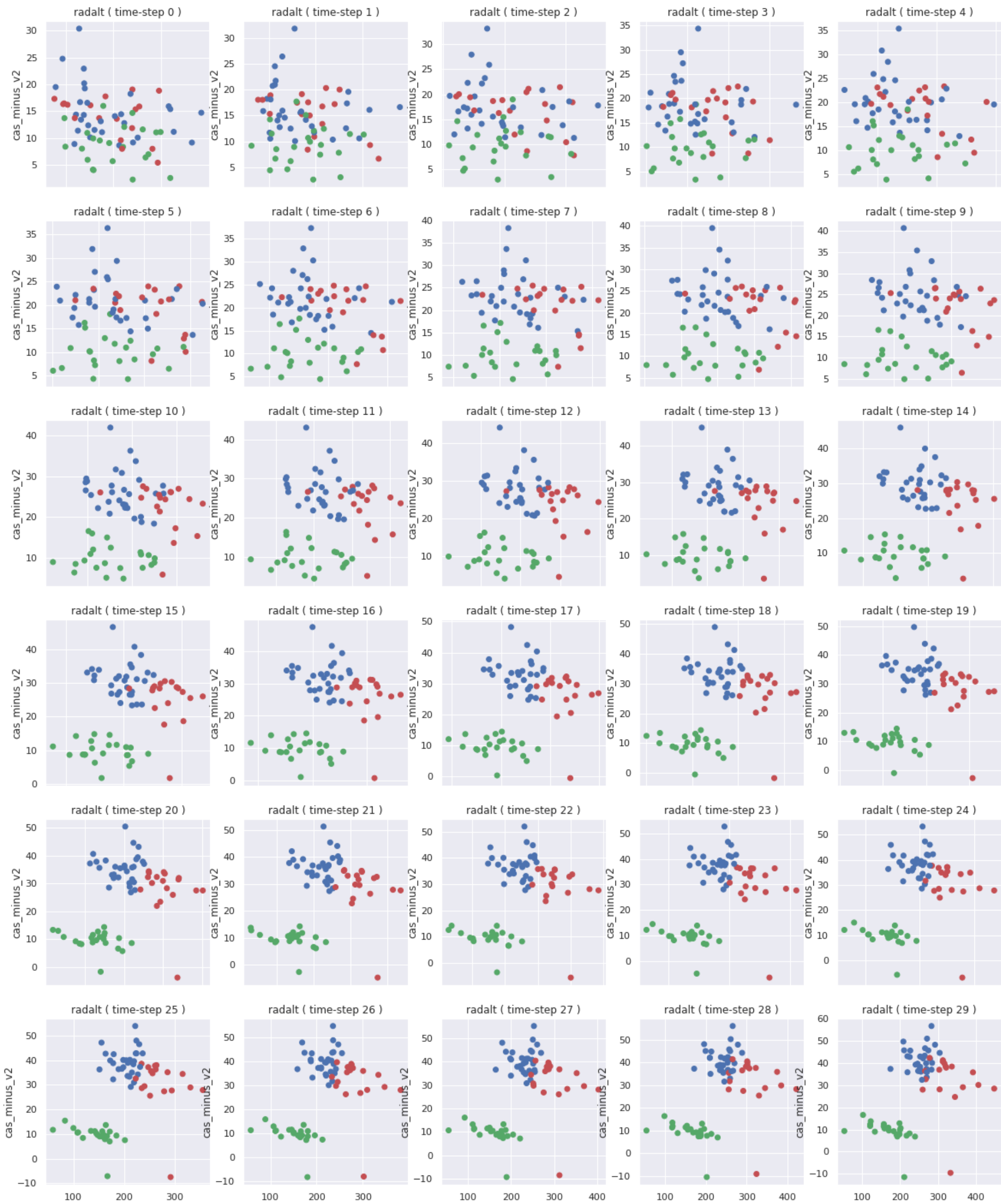


Figure 20 Flights clusters scatter plot at all time steps (Airspeed vs Altitude)

As Figure 20 demonstrates, around the liftoff event the clusters are mixed and while the time-steps are increased the clusters are separated from each other on the main diagonal. This is an indication of a positive correlation between flights' airspeed and altitude and similar order of contribution in terms of classifying the clusters. More specifically, the green cluster is separated from blue and red clusters and is associated to flights with low altitude and airspeed. Blue and red clusters are flights with higher airspeed and altitude. Their airspeed is more or less similar (blue flights have a little higher airspeed than red flights) but their difference is more meaningful in terms of their altitude where red flight has higher altitude than blue flights. The mentioned interpretations are also shown in the following figures where plots of each parameter vs time-steps are presented.

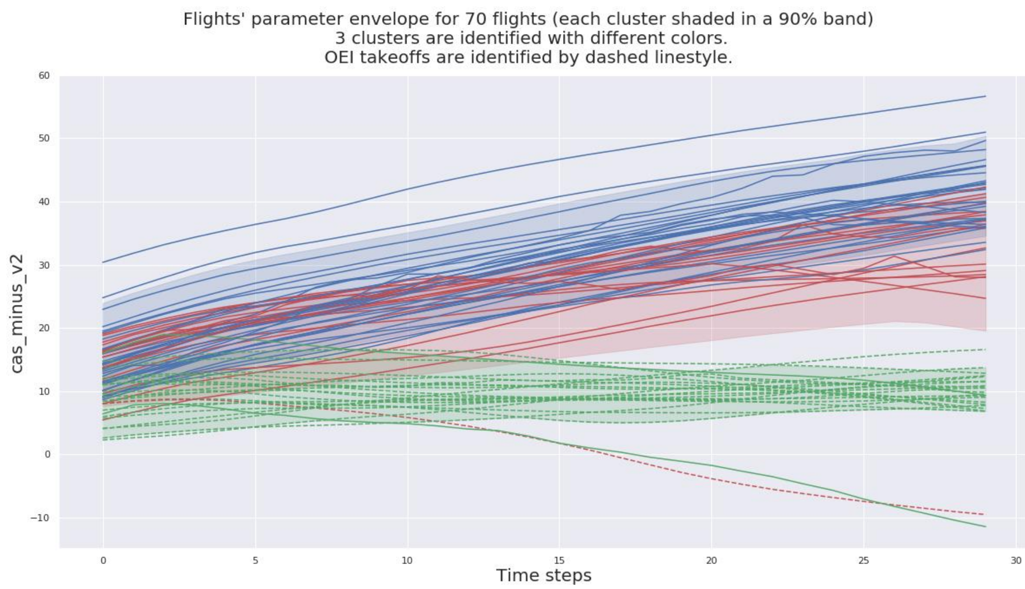


Figure 21 Airspeed (Flights Clusters) – GMM used Airspeed and Altitude

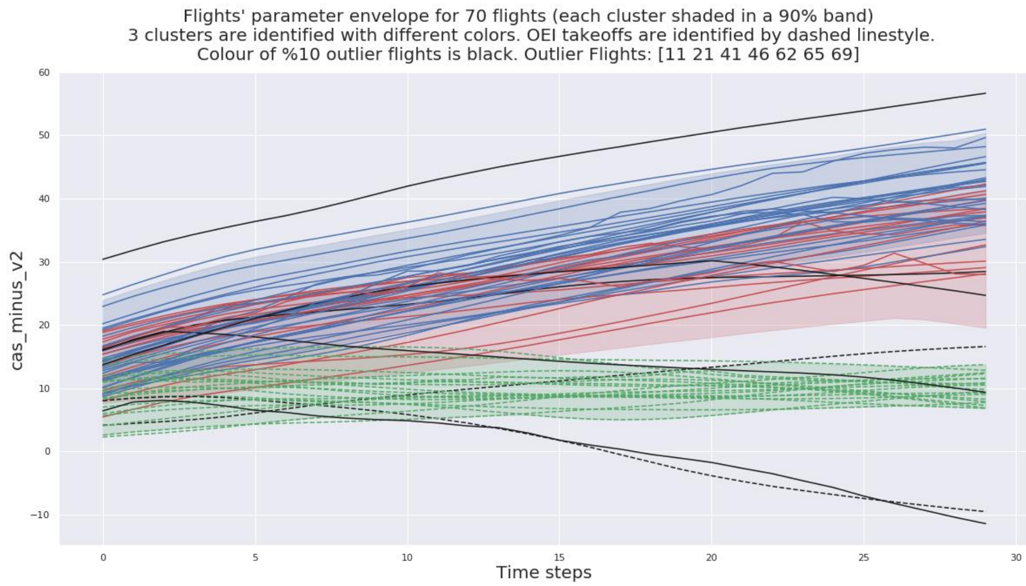


Figure 22 Airspeed (Flights Outliers) – GMM used Airspeed and Altitude

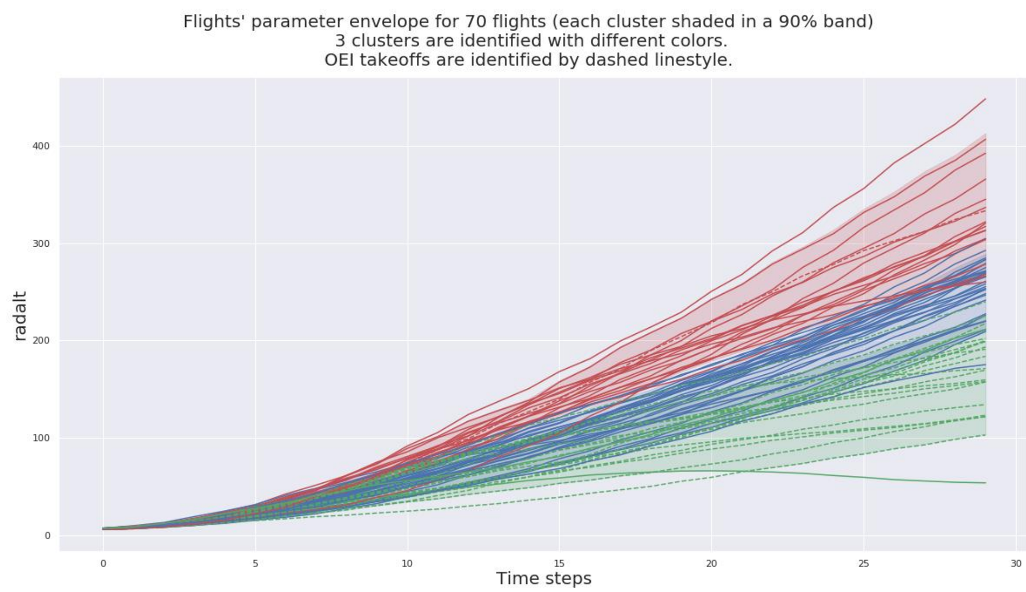


Figure 23 Altitude (Flights Clusters) – GMM used Airspeed and Altitude

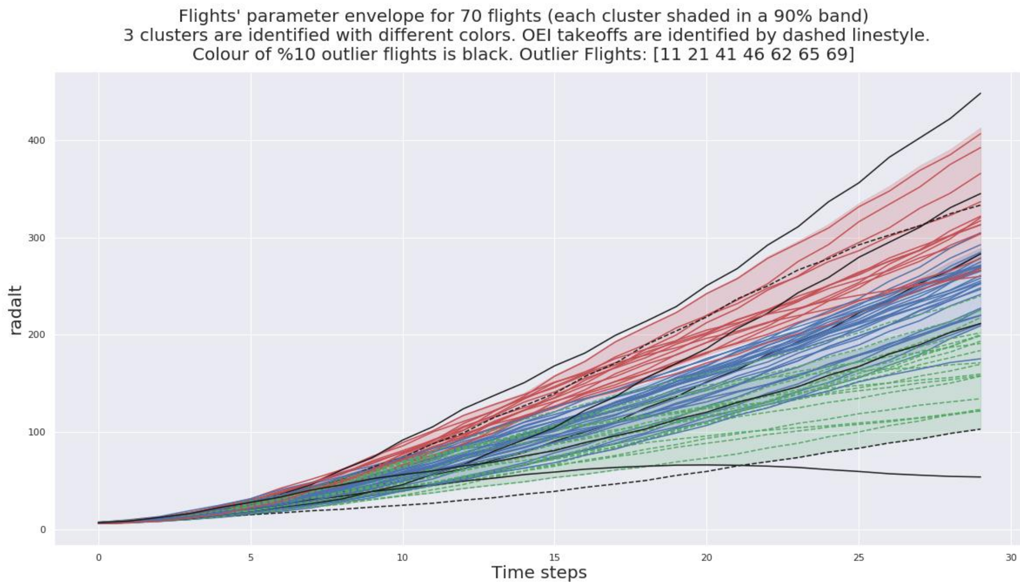


Figure 24 Altitude (Flights Outliers) – GMM used Airspeed and Altitude

Interpretation of the above plots are like what was discussed for the airspeed and altitude's discussion individually. Note that OEI flights mostly gathered in one cluster in both altitude and airspeed plots

Airspeed, Altitude and Pitch

For three parameters, first a probability density histogram plot for each parameter is presented.

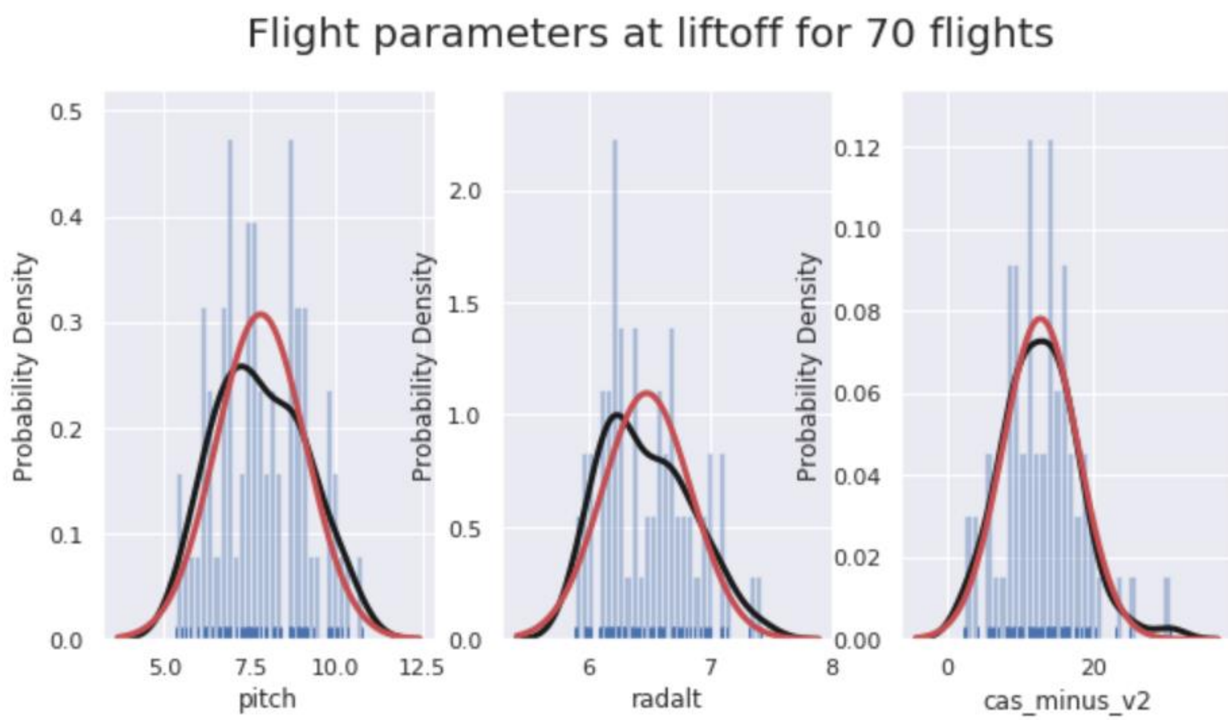


Figure 25 Airspeed, Altitude and Pitch (Flights Optimum Number of Cluster)

Distribution of each parameter's probability density function follow a normal distribution.

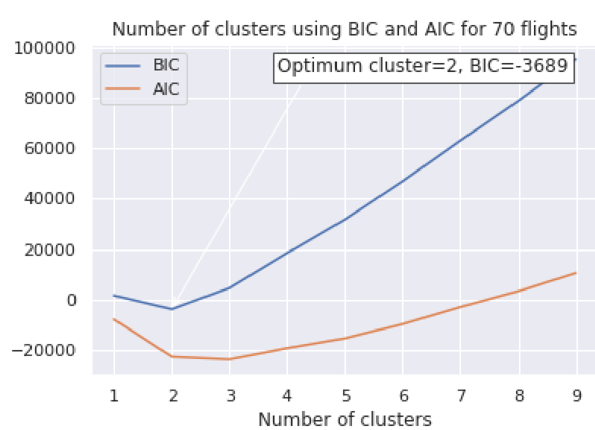


Figure 26 Airspeed, Altitude and Pitch (Flights Optimum Number of Cluster)

BIC determined 2 clusters. In the following, three scatter plots for each two parameters at each time-steps are presented.

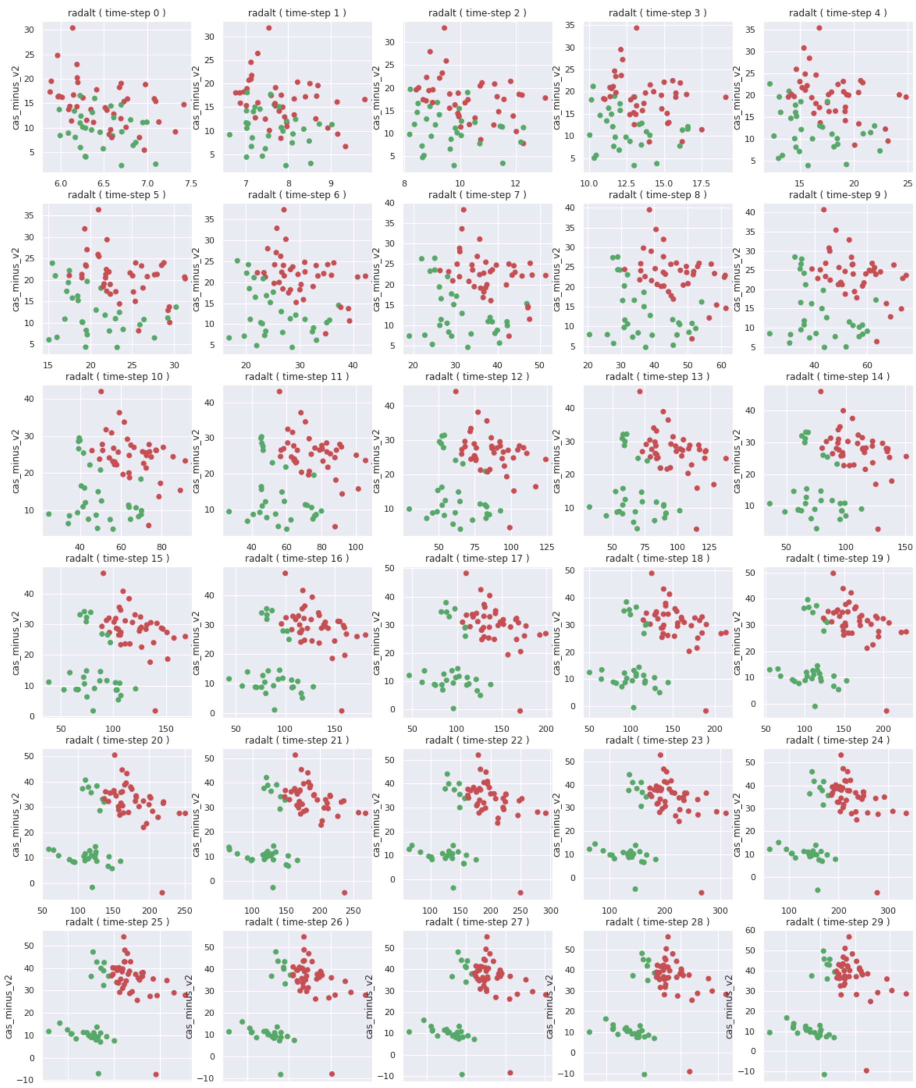


Figure 27 Flights clusters scatter plot at all time steps (Airspeed vs Altitude)

Figure 27 is like Figure 20 except that the added pitch parameter caused a change in the BIC result from 3 to 2 clusters. Looking at flights located at the top right corner of latest time-steps, it is obvious that those flights can be classified into two separate clusters, as was shown on Figure 20.

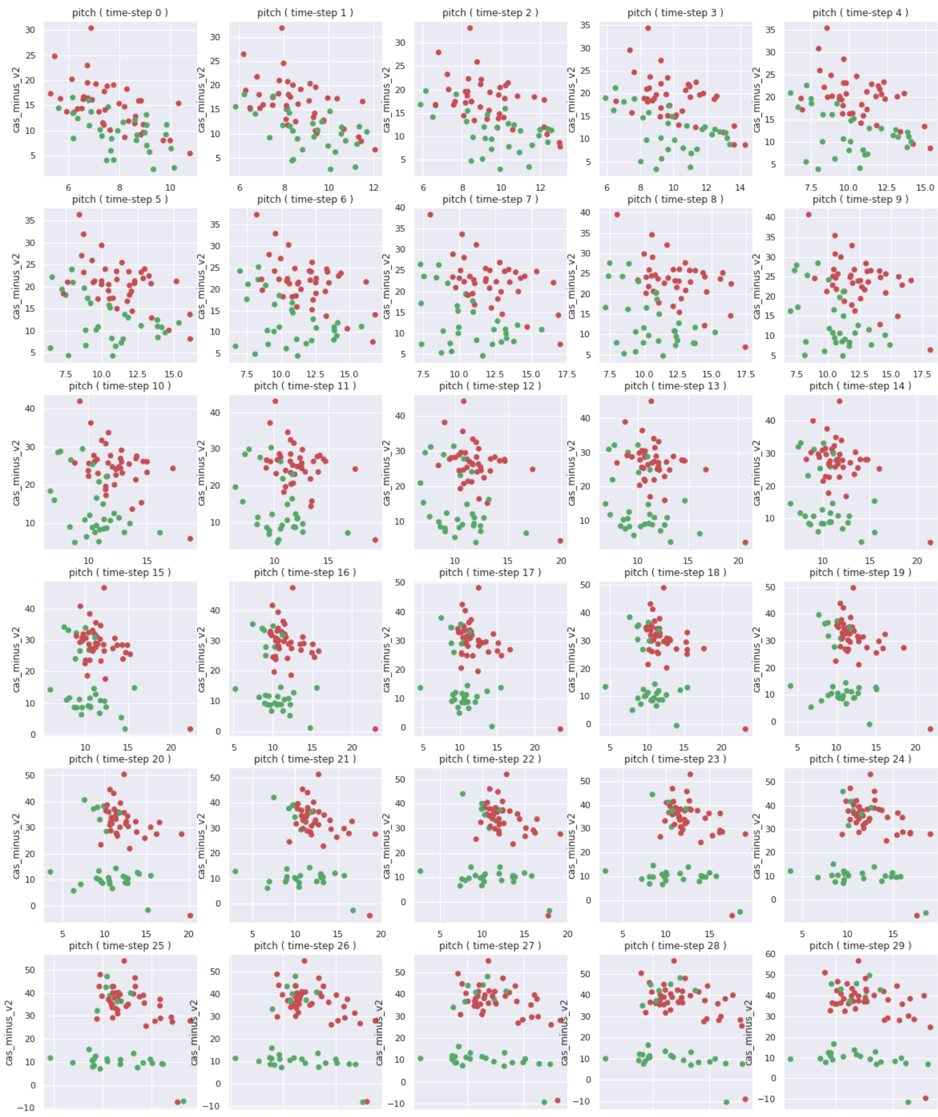


Figure 28 Flights clusters scatter plot at all time steps (Airspeed vs Pitch)

Vertical separation of clusters, illustrated on Figure 28, indicates that airspeed has more power than pitch in terms of separating the clusters. In a separate GMM analysis with 3 clusters, it was noticed that mixed green and red flights were separated into two different clusters.

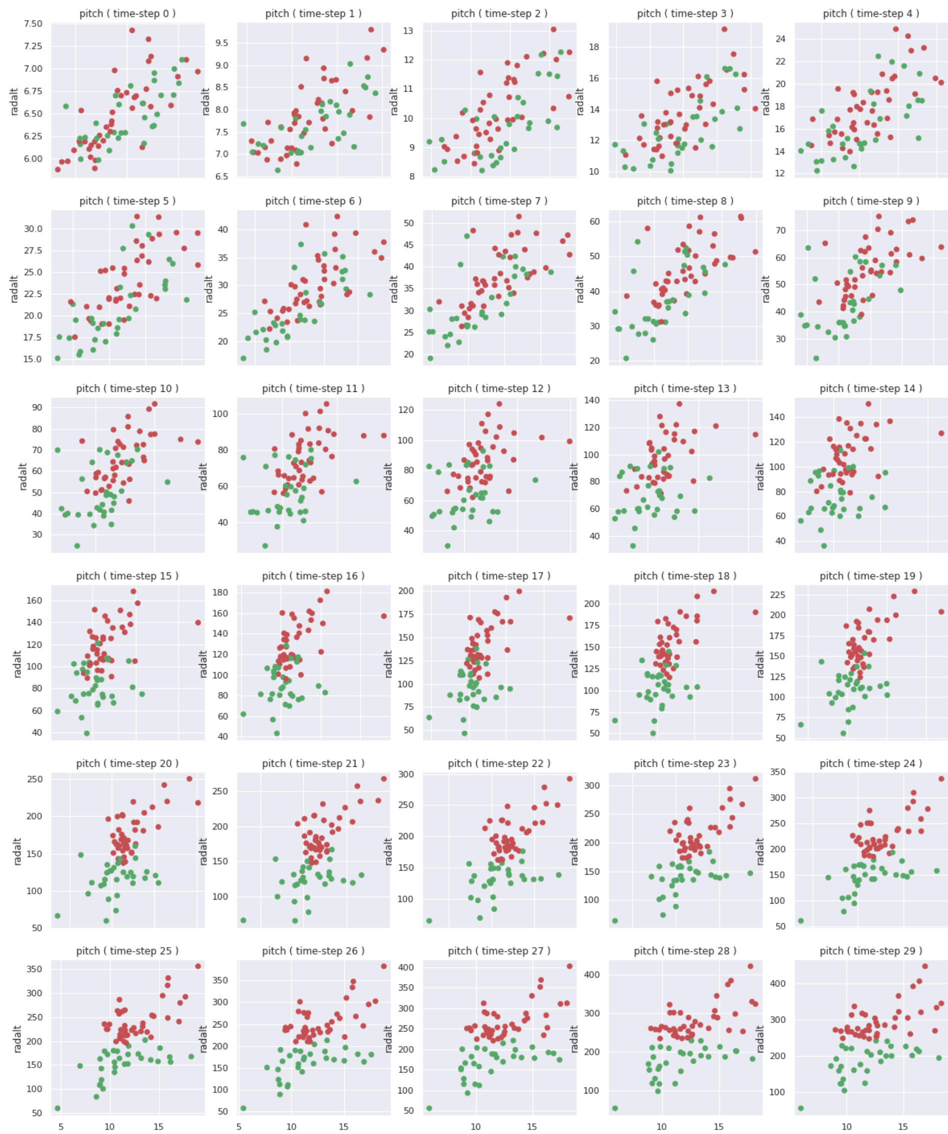


Figure 29 Flights clusters scatter plot at all time steps (Pitch vs altitude)

Vertical separation of clusters, illustrated on Figure 29, indicates that altitude has more power than pitch in terms of separating the clusters. Altogether considering all the plots, both airspeed and altitude have similar strength and higher than pitch in terms of separating the flights clusters.

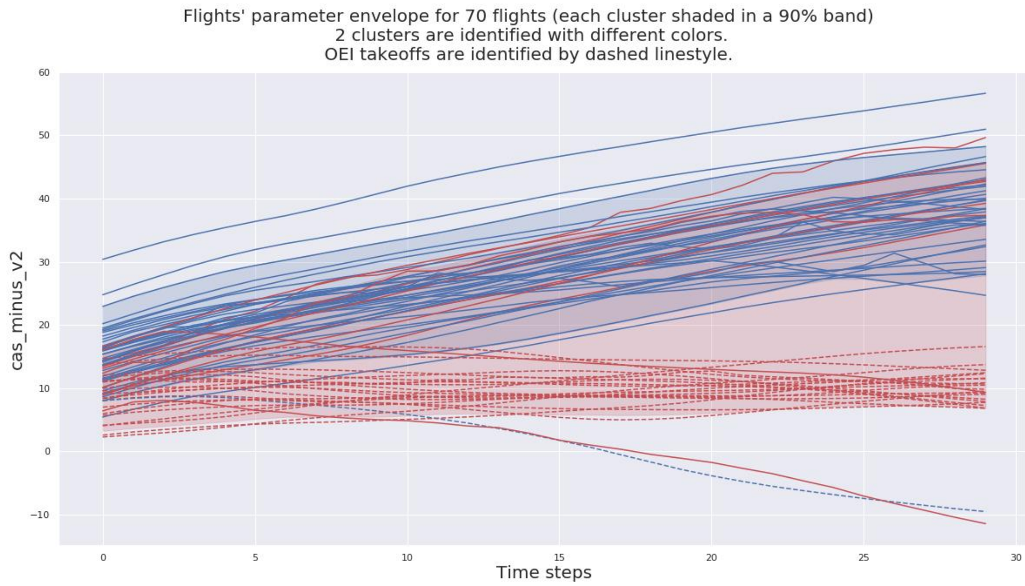


Figure 30 Airspeed (Flights Clusters) – GMM used Airspeed, Altitude and Pitch

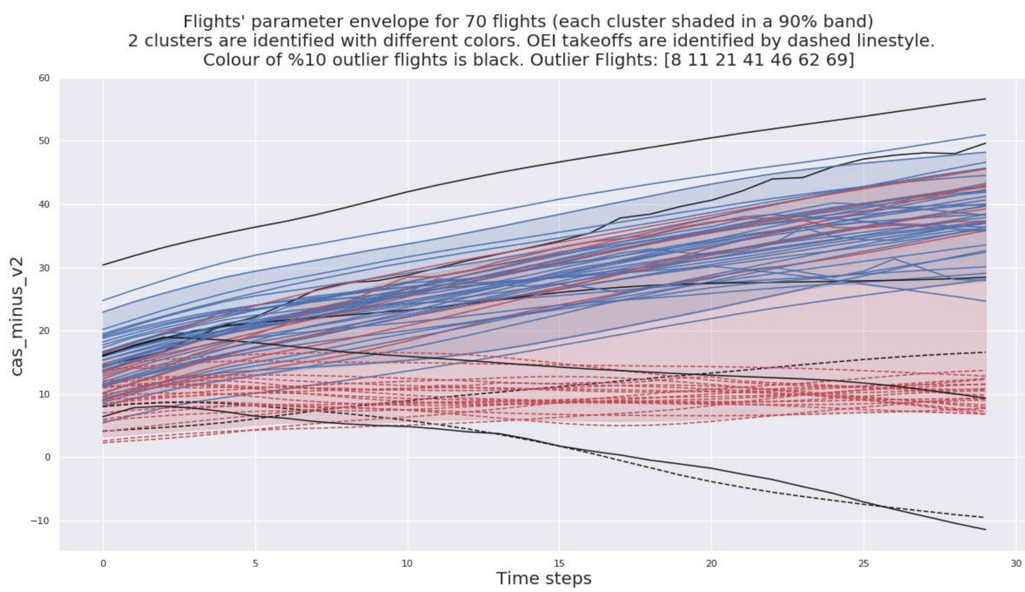


Figure 31 Airspeed (Flights Outliers) – GMM used Airspeed, Altitude and Pitch

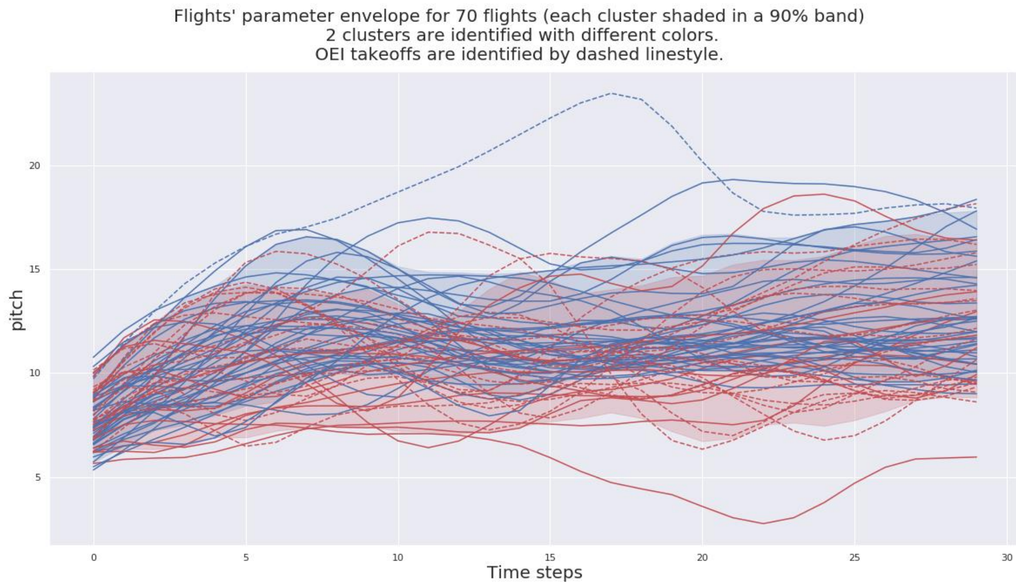


Figure 32 Pitch (Flights Clusters) – GMM used Airspeed, Altitude and Pitch

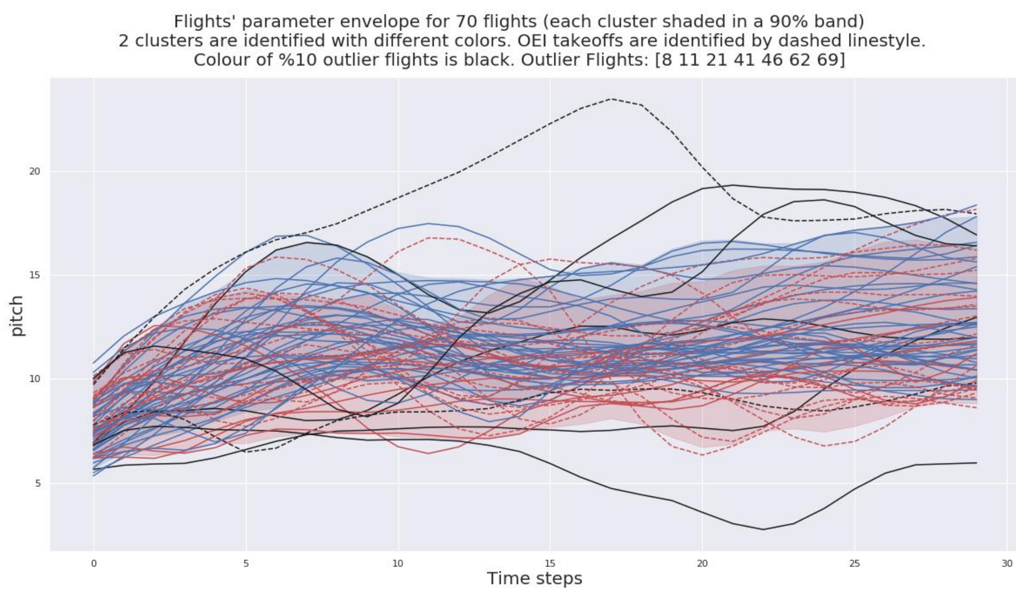


Figure 33 Pitch (Flights Outliers) – GMM used Airspeed, Altitude and Pitch

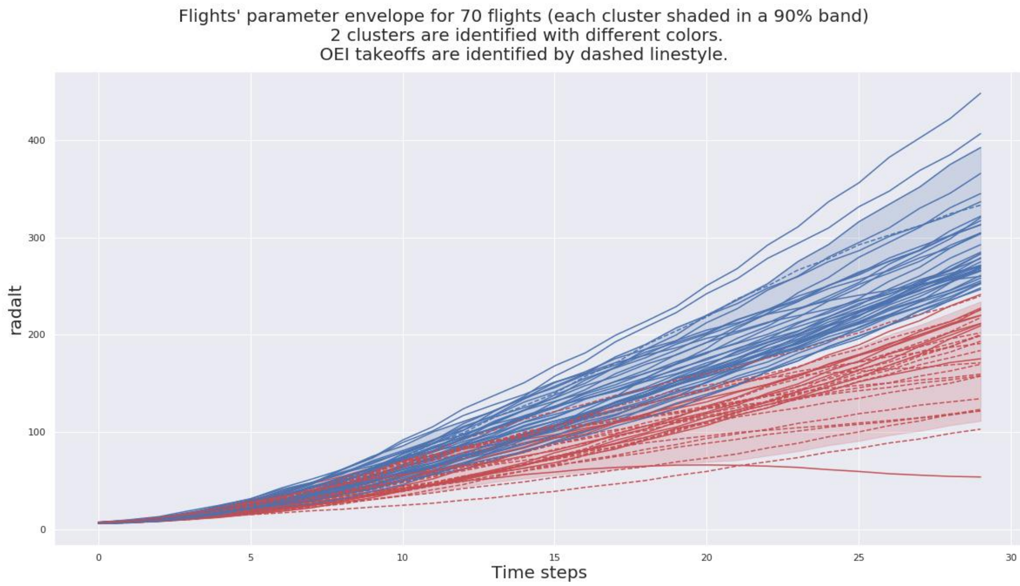


Figure 34 Altitude (Flights Clusters) – GMM used Airspeed, Altitude and Pitch

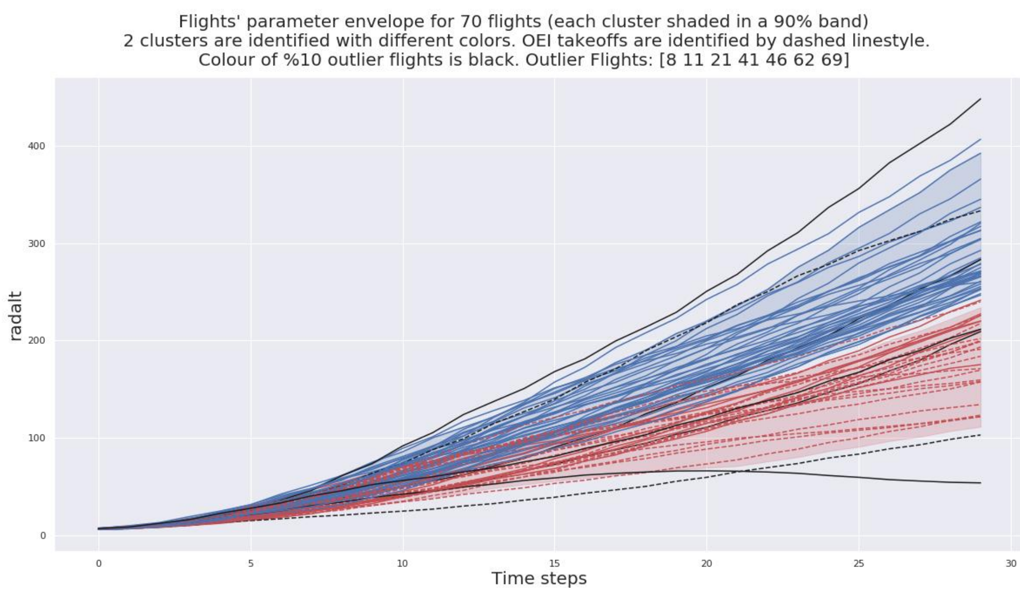


Figure 35 Altitude (Flights Outliers) – GMM used Airspeed, Altitude and Pitch

Plots of the flights clusters show more meaningful results for airspeed and altitude than pitch. Majority of the OEI flights (dash style lines) have been detected in the low altitude and low airspeed cluster (red color lines).

Takeoff Maneuver (475 time-steps)

Flap Lever Position, Airspeed and Altitude

In this section, results of GMM analysis performed on three parameters (flap lever position, altitude, airspeed) are presented. Number of GMM attributes (time-steps) in this section is 475 (length of the longest flight).

Results in the following plots are based on 3 clusters. BIC proposed 8 clusters when Flap Lever Position discrete parameter was added to the airspeed and altitude quantitative parameters. However, since the interpretation of results based on 8 clusters was considered verbose, 3 clusters was used. This number of clusters was determined by BIC when only airspeed and altitude were used in the GMM analysis (refer to Figure 19). Note that while BIC presents the optimal number of clusters, during this research it was noticed that the specialists' judgment to propose a meaningful and interpretable number of clusters is more important that what BIC suggests.

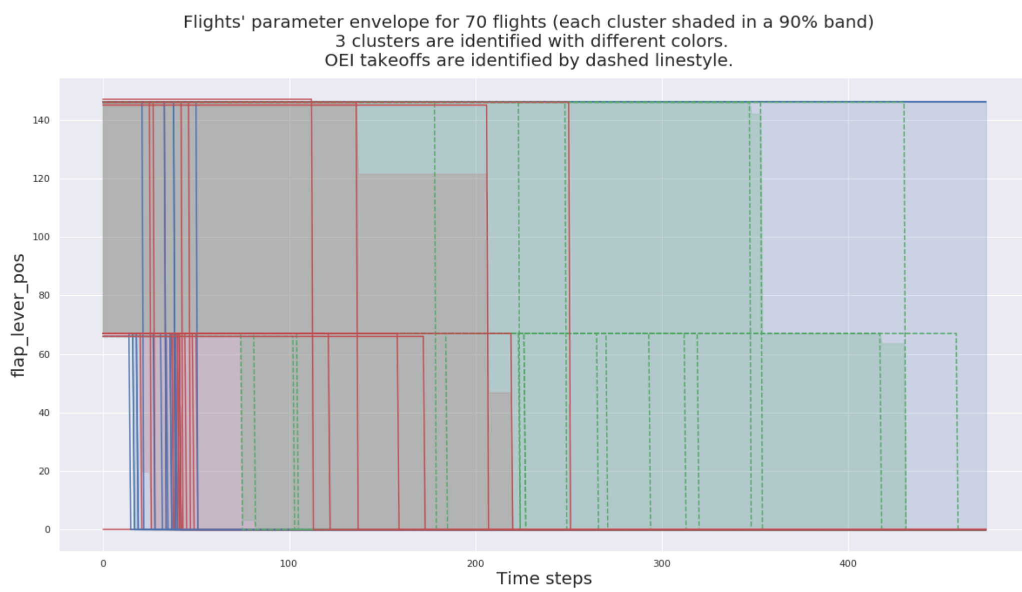


Figure 36 Flap Lever (Flights Clusters) – GMM used Flap Lever, Airspeed and Altitude

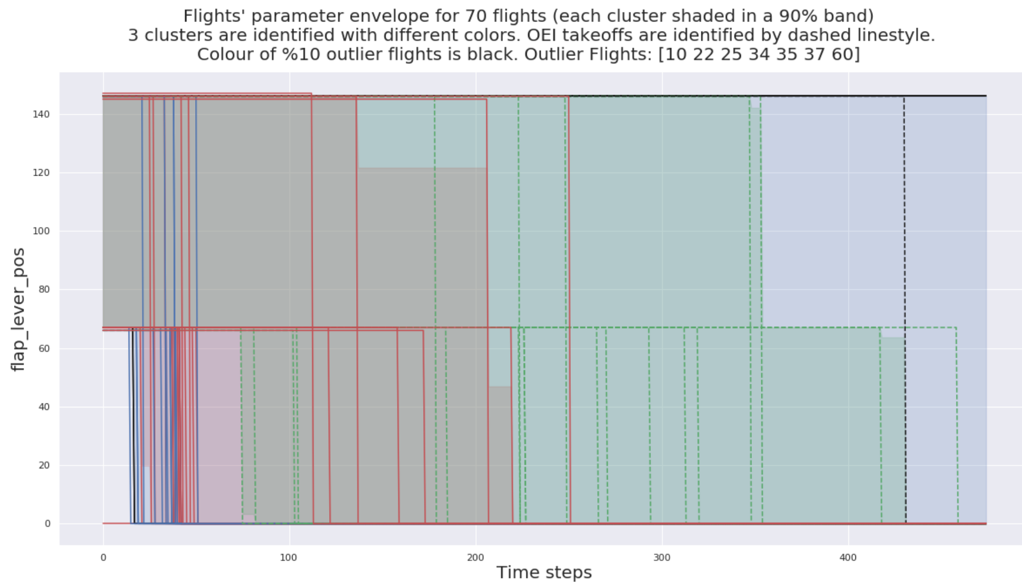


Figure 37 Flap Lever (Flights Outliers) – GMM used Flap Lever, Airspeed and Altitude

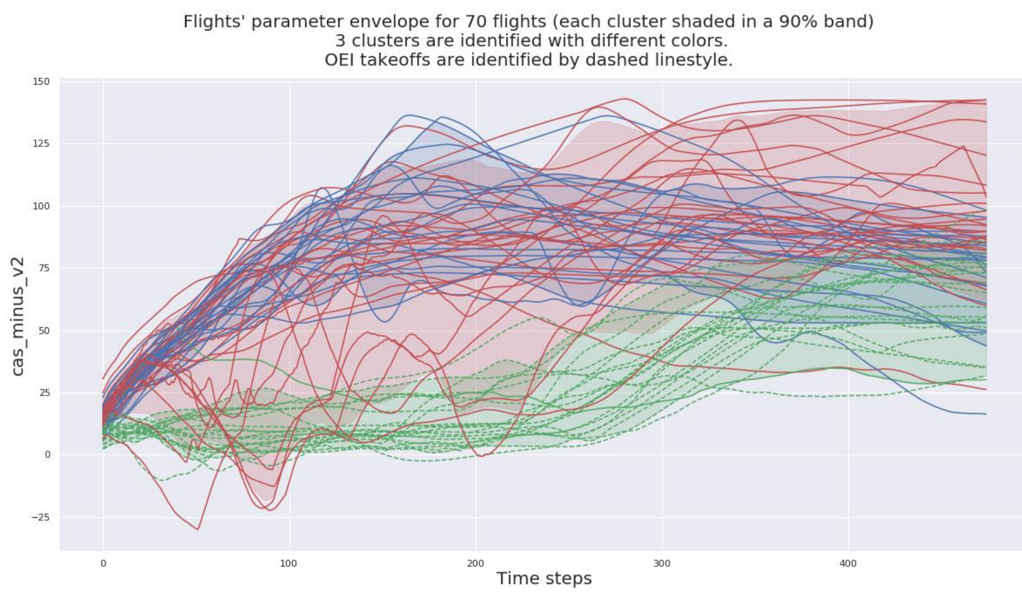


Figure 38 Airspeed (Flights Clusters) – GMM used Flap Lever, Airspeed and Altitude

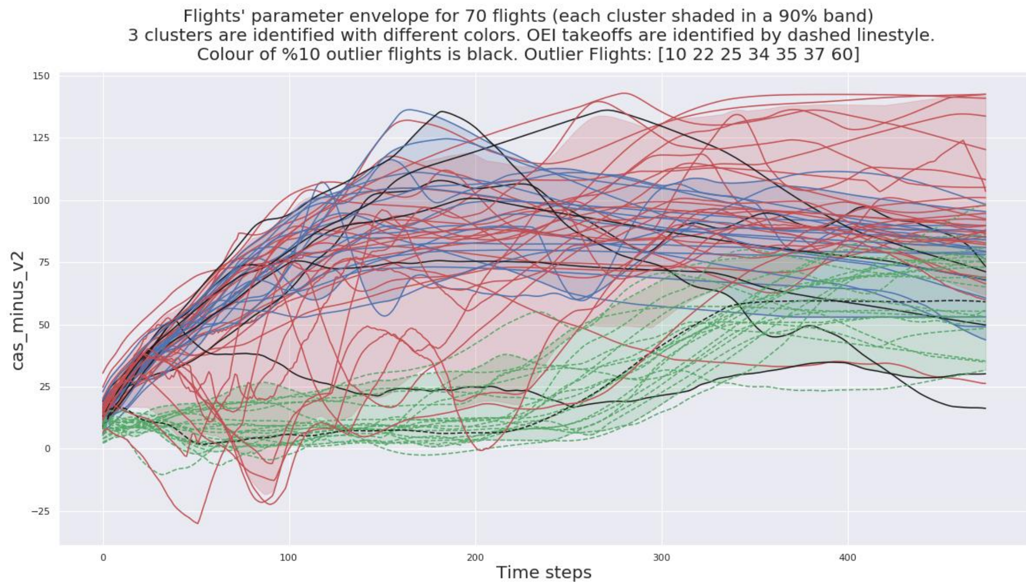


Figure 39 Airspeed (Flights Outliers) – GMM used Flap Lever, Airspeed and Altitude

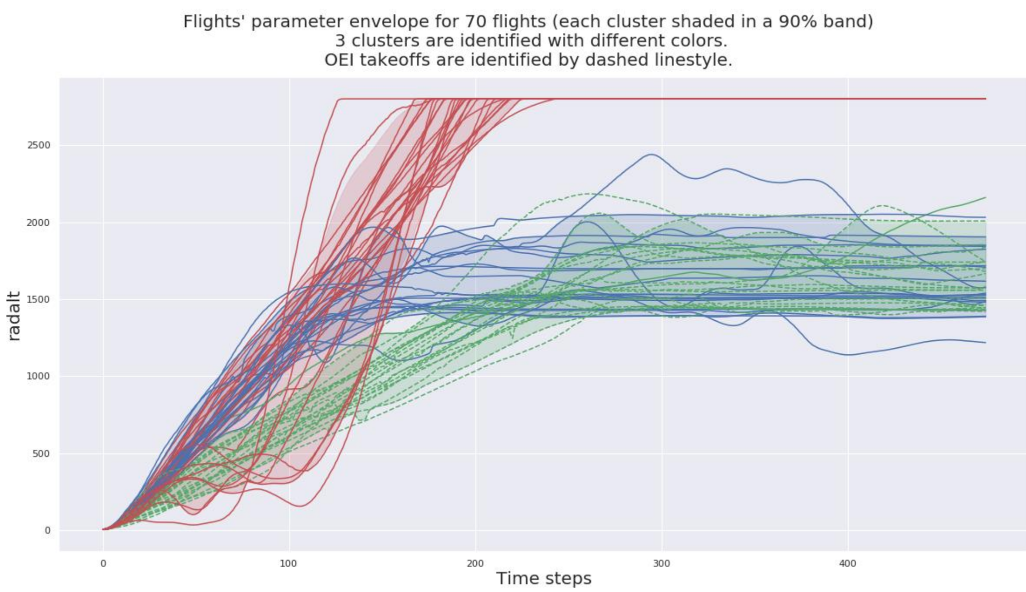


Figure 40 Altitude (Flights Clusters) – GMM used Flap Lever, Airspeed and Altitude

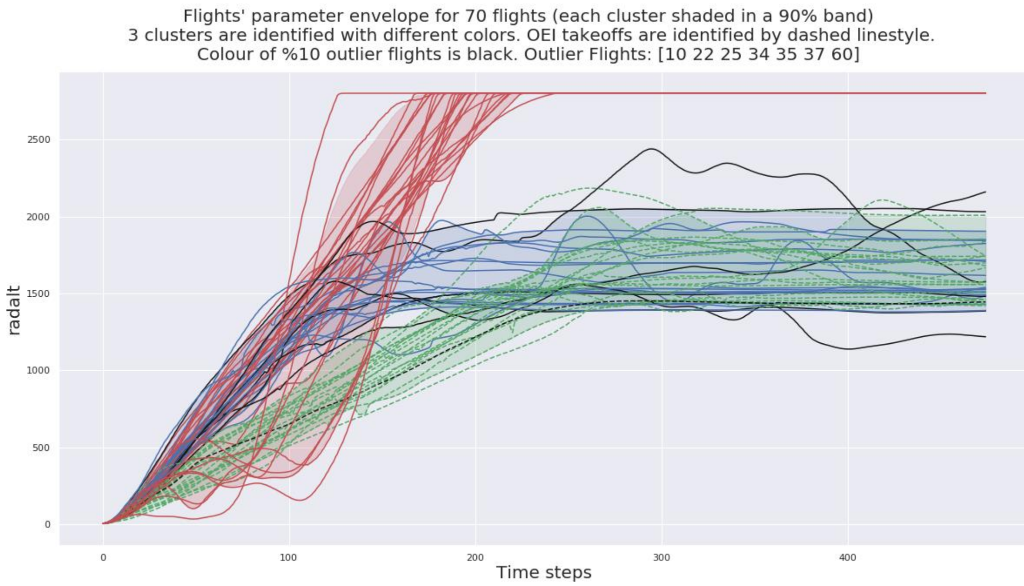


Figure 41 Altitude (Flights Outliers) – GMM used Flap Lever, Airspeed and Altitude

Interpretation of the above plots are like what was discussed for the airspeed and altitude's discussion. Note that OEI flights mostly gathered in one cluster identified with green color.

Takeoff Event – Flap Retraction

The following two plots have been prepared to have a better understanding about the flap retraction (or cleanup) event distribution. For Figure 42, GMM has used the Airspeed and Altitude values from the liftoff to the longest takeoff activity. The flap retraction event has been pointed by dots on the trajectory clusters.

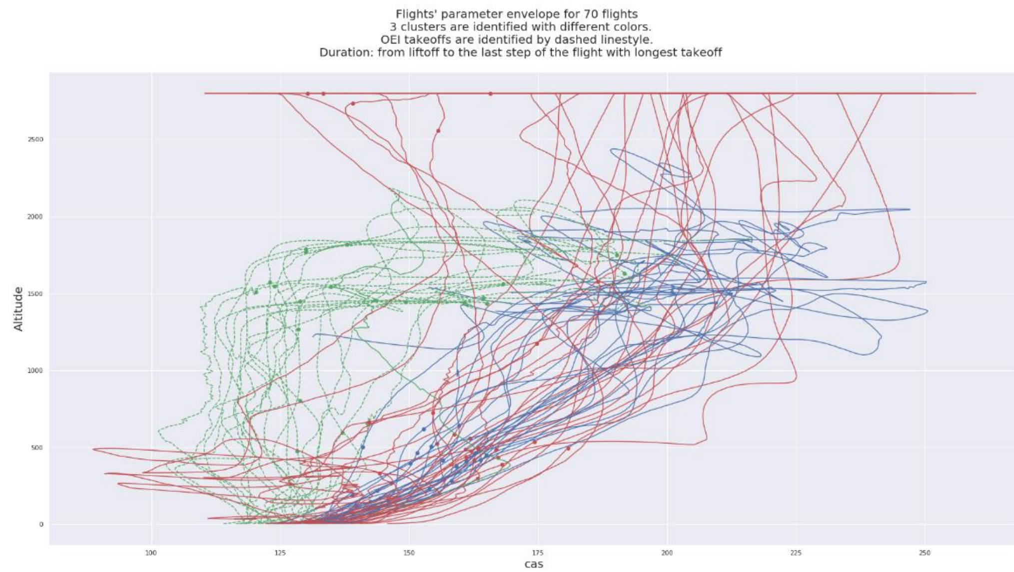


Figure 42 Altitude vs Airspeed (Flights Clusters) – GMM used Flap, Airspeed and Altitude

It seems that 3 cluster is a fair representative number of clusters because as Figure 42 illustrates, the flights are separated into 3 clusters: (1) green dashed curves, OEI flights with low altitude, (2) blue curves, “not OEI” flights with low altitude, and (3) red curves, flights with high altitude. Of course, the decision about the best representative number of clusters depends on the pilot trainers/specialists’ judgement about the interpretability of the results.

For Figure 43, GMM has only used the Airspeed and Altitude values at the flap cleanup event (one time-step). On that plot, OEI Takeoffs are identified by an X on top of the dots.

Figure 43 Altitude vs Airspeed (Flights Clusters) – GMM used Flap, Airspeed and Altitude

Figure 43 demonstrates the GMM sensitivity to the input data. This study demonstrates the difference between time-series clustering and event clustering. This event-based clustering can reach to outliers' identification without using a Log Probability Density Method (explained on page 37). Figure 44 is a reproduction of Figure 43 with added explanations, confirmed and explained by a Paladin AI specialist, indicating detected anomaly flights.

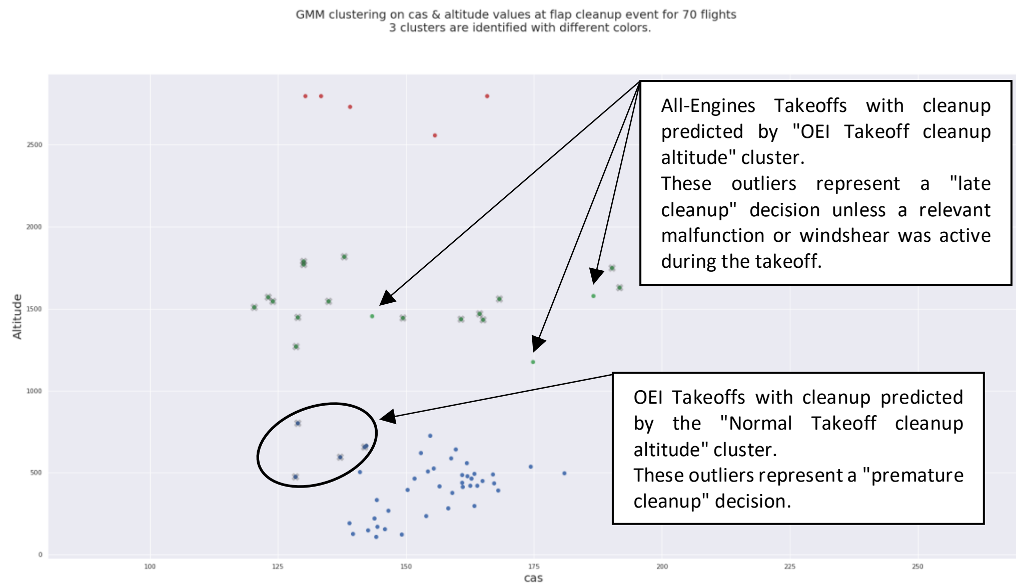


Figure 44 Altitude vs Airspeed (Flights Outlier) – GMM used Flap, Airspeed and Altitude

Landing Flare Maneuver

The goal is to run GMM on Airspeed and Altitude during the landing flare maneuver. Landing flare activity starts at 50 ft altitude during the aircraft landing approach. GMM input is a 2d-array where samples (rows) are 91 flights and attributes (columns) are 50 time-steps from touchdown. The flights parameters vectors have different lengths and need to be adjusted and fitted to one size vector (here 50 time-steps length). For every flight, the Airspeed and Altitude values need to be resampled (interpolated) for 50 equal time-steps from radio altitude 50 to touchdown decremented by 1 time-step. Refer to page 32 for an explanation about the used interpolation method.

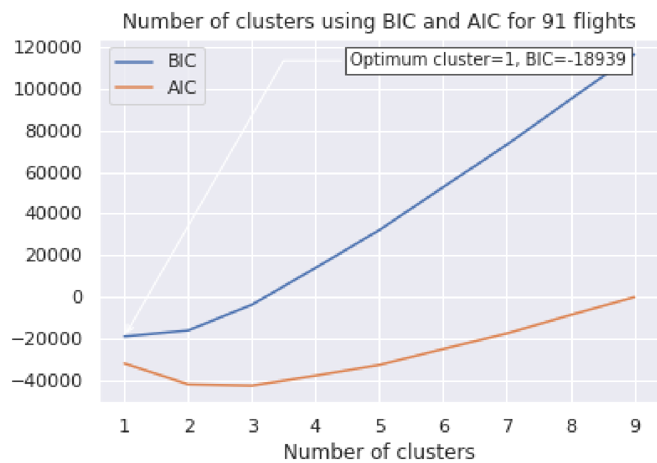


Figure 45 Airspeed and Altitude (Flights Optimum Number of Cluster)

Since the BIC difference between 1 and 2 clusters in Figure 45 is minor, GMM analysis is performed for 2 clusters. Figure 20Figure 46 demonstrates flights airspeed-altitude scatter plots at all time-steps.

Figure 46 Flights clusters scatter plot at all time steps (Airspeed vs Altitude)

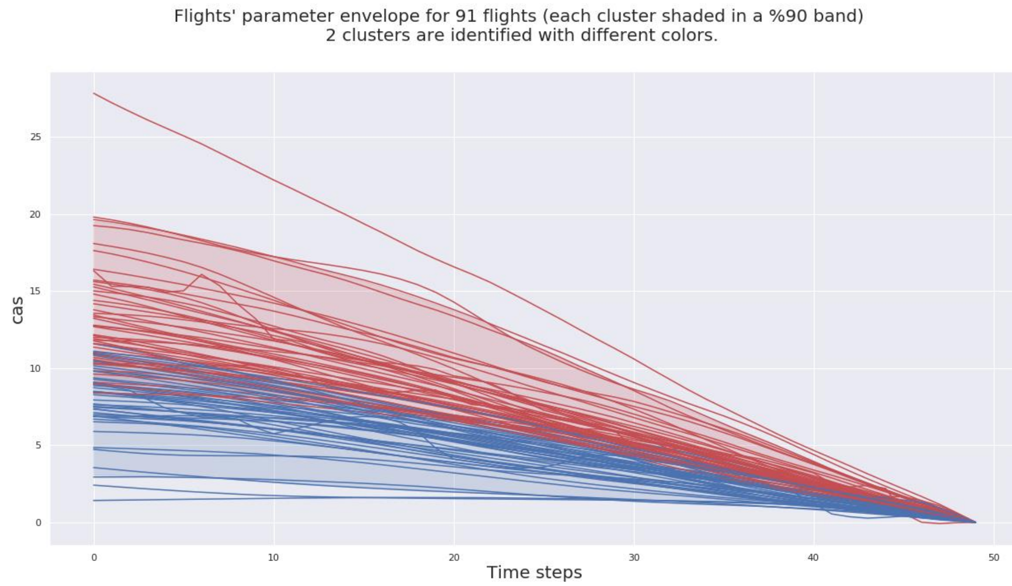


Figure 47 Airspeed (Flights Clusters) – GMM used Airspeed and Altitude

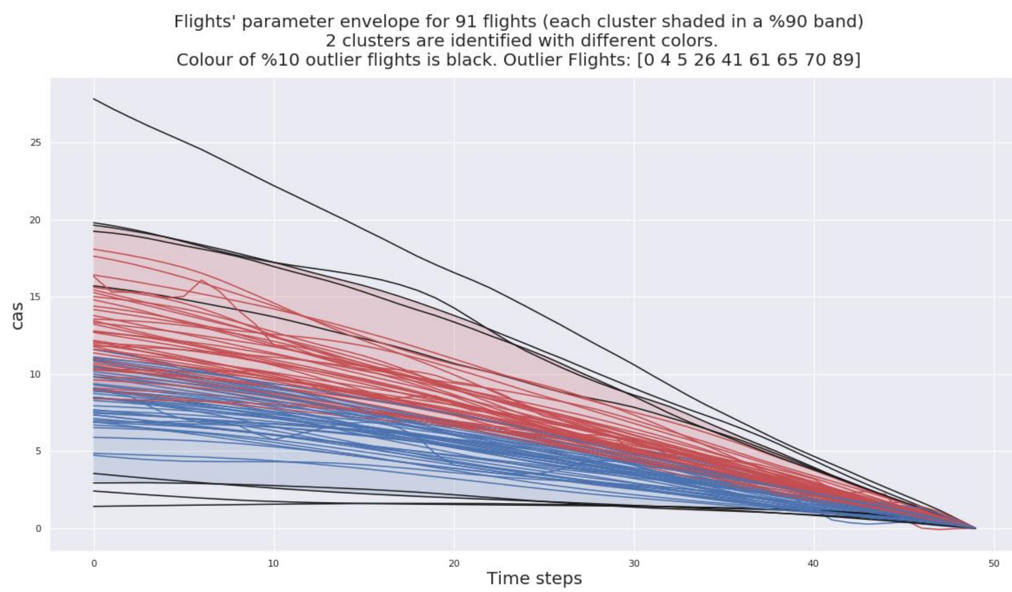


Figure 48 Airspeed (Flights Outliers) – GMM used Airspeed and Altitude

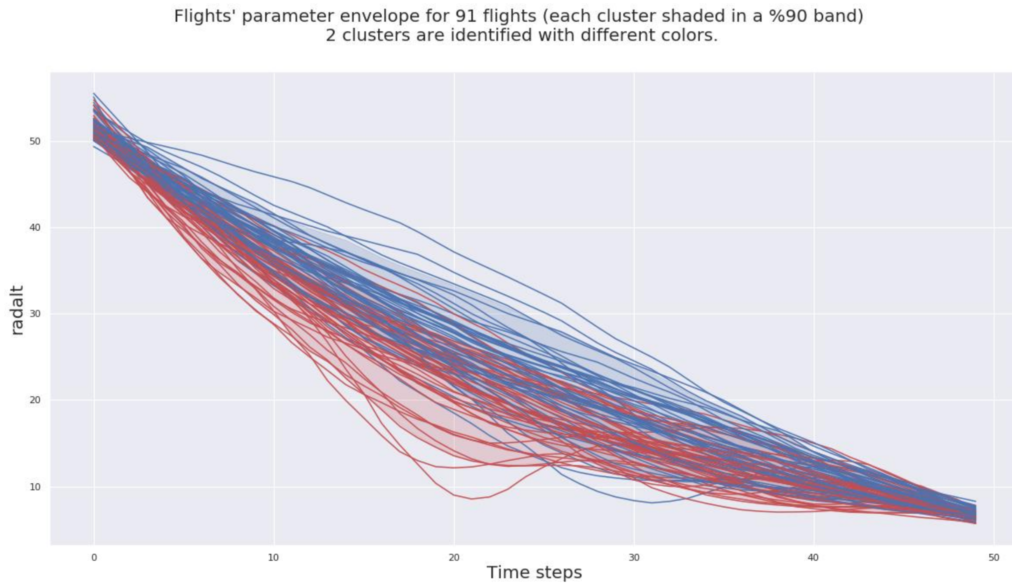


Figure 49 Altitude (Flights Clusters) – GMM used Airspeed and Altitude

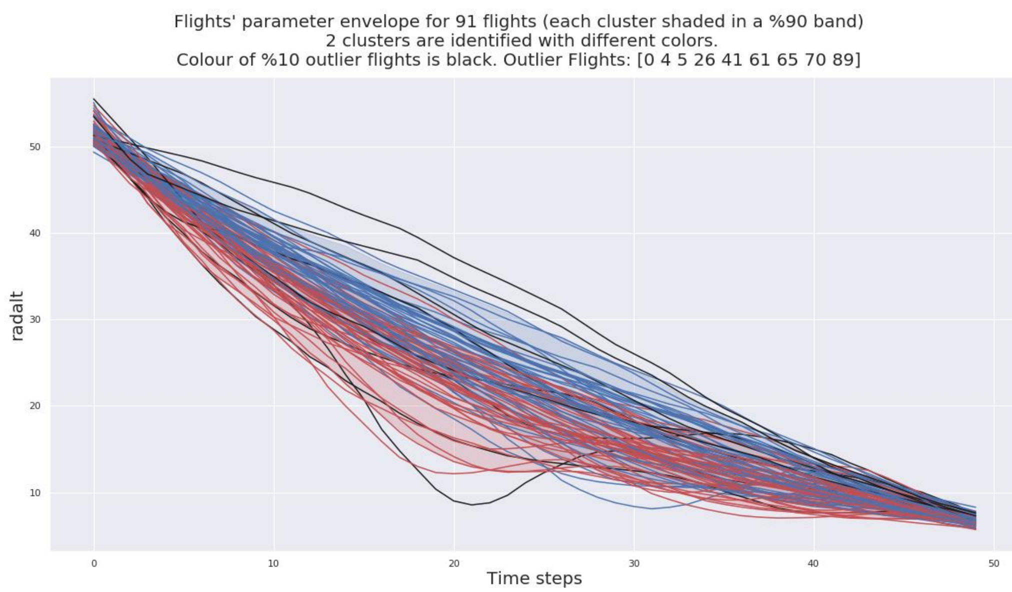


Figure 50 Altitude (Flights Outliers) – GMM used Airspeed and Altitude

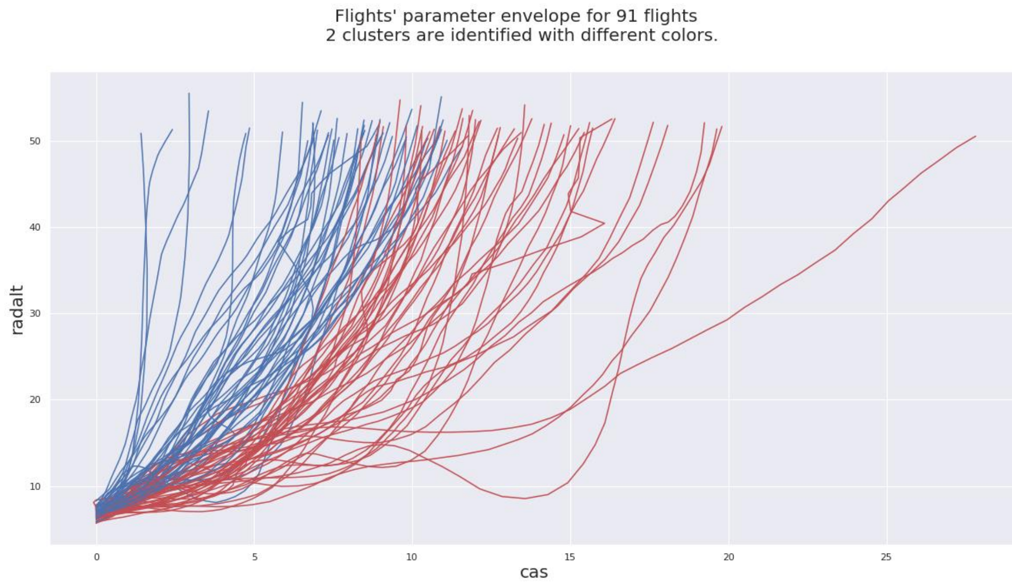


Figure 51 Altitude vs Airspeed (Flights Clusters) – GMM used Airspeed and Altitude

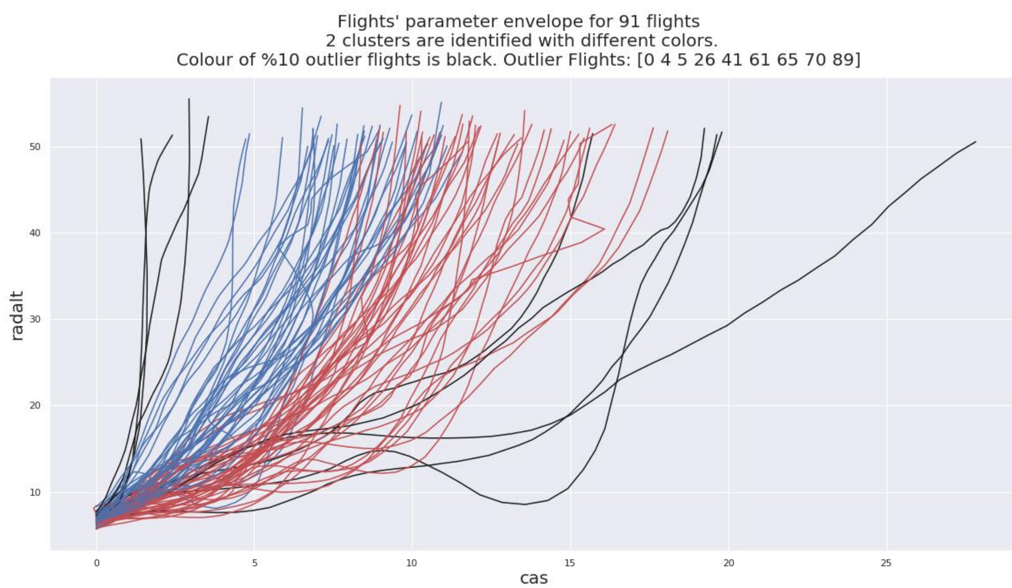


Figure 52 Altitude vs Airspeed (Flights Outliers) – GMM used Airspeed and Altitude

According to above depicted figures for two clusters, anomaly flights are meaningful and different from most flights in each cluster.

Conclusion

This internship research started with a generic literature review. After introducing some core concepts on clustering algorithm for unsupervised learning, an overview on the theory behind Gaussian Mixture Model (GMM) was presented. Overviews on the flight's anomaly detection methods as well as different pilot training paradigms were given and concluded that GMM can be an effective analytics technique for inferring pilot competency in a data-driven evidence-based and adaptive environment. GMM was used in this research for clustering and anomaly detection of pilot training data.

In the Methodology chapter, the investigated flight's activities and event were pictured and a pipeline from dataset to GMM input was designed. Two sample's comparison and alignment methods (necessary for GMM analysis), were explained and it was concluded to use "same time-step method" and "aligned time-step method" with a linear interpolation approach. For outlier detection, a "probability averaging method" was proposed and experimented against "log-probability density method" and it was concluded to use the latter.

In the results and conclusion chapter, results for takeoff and landing flare maneuvers as well as flap retraction event were presented. It was observed that

(1) GMM can identify clusters and detect anomaly among flights at both maneuver and event levels. For example, OEI flights could be automatically detected. Examples are all over the report. Refer to Figure 12 or Figure 52 as typical examples.

(2) Different time-varying pilot control strategies appear to be distinguished by different clusters even if the clusters cross during a given time window. For example, GMM clustering result for pitch parameter, represented pilots' separate patterns of maneuver's behaviour (refer to Figure 15)Figure 12. Another example, is Flap Retraction event clustering, explained under Figure 42.

(3) While BIC method in GMM determines the optimal number of clusters and it is a good starting point, the specialists' judgment to propose a meaningful and interpretable number of clusters is more important that what BIC suggests. As typical examples, refer to page 55 and an explanation under Figure 42.

(4) The event-based clustering can reach to outliers' identification without using a "Log Probability Density Method" from GMM. For example, location of OEI flights identified on Figure 44, represent pre-mature or late flap retractions.

Future Works

The GMM method can be tested on other types of flight's activities. The pilot training data can be evaluated by more advance time-series algorithms such as LSTM as a deep neural network recurrent method for clustering. Reinforcement learning algorithm can also be used where it can expose a virtual pilot to a virtual environment and provide it with human pilots' flight data and a goal.

The approach used in this research, can be implemented to any domain with possibility of time-series data collection and transformation such as drivers' training and behaviors analysis, or interactive online educational trainings.

Bibliography

Das, S., Matthews, B., Srivastava, A., & Oza, N. (2010). Multiple kernel learning for heterogeneous anomaly detection: algorithm and aviation safety case study. *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 47–56). ACM, Washington, DC: http://dx.doi.org/10.1145/1835804.1835813.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the EM algorithm. *J. R. Stat. Soc. 39*, 1-38.

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, Vol. 96, No. 34* (pp. 226–231). Portland, OR: AAAI Press.

From Wikipedia, t. f. (2020). *DBSCAN.*

Ganti, V., Ramakrishnan, R., Gehrke, J., Powell, A., & French, a. J. (1999). Clustering large datasets in arbitrary metric spaces. *n Proceedings of the 15*, (pp. 502-511). Sydney.

Guha, S., Rastogi, R., & Shim, K. (2001). CURE: An Efficient Clustering Algorithm for Large Databases. *Information Systems*, 35-58.

Hartigan, J., & Wong, M. (1979). A k-means clustering algorithm. *Applied Statistics*, 28, 100-108.

IATA. (2013). *Evidence-Based Training Implementation Guide.* Montreal - Geneva: International Air Transport Association.

ICAO. (2014). Aviation Training Going Forward. *ICAO Training Report, Vol. 4, No. 2*.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2017). *An Introduction to Statistical Learning.* Springer.

Klassen, M. (2020, Feb 27). How a Data-Driven Approach to Pilot Training will make Aviation Safer while Saving Airlines Money. Canada.

Leskovec, J., Rajaraman, A., & Ullman, J. (2014). *Mining of Massive Datasets.*

Li, L., & Hansman, R. J. (2013). *Anomaly Detection In Airline Routine Operations Using Flight Data Recorder Data.* Cambridge, MA 02139 USA: MIT International Center for Air Transportation (ICAT), Report No. ICAT-2013-4.

Li, L., Das, S., Hansman, R. J., Palacios, R., & Srivastava, A. N. (2015). Analysis of Flight Data Using Clustering Techniques for Detecting Abnormal Operations. *JOURNAL OF AEROSPACE INFORMATION SYSTEMS*.

Li, L., Hansman, R. J., Palacios, R., & Welsch, R. (2016). Anomaly detection via a gaussian mixture model for flight operation and safety monitoring. *Transportation Research Part C 64*, 45-57.

Reynolds, D. A. (2008). Gaussian Mixture Models. *Encyclopedia of Biometric Recognition Springer, (2)*, 659–663.

Salvador, S., & Chan, P. (2004). Toward accurate dynamic time warping in linear time and space. *Intelligent Data Analysis, 11*, 70-80.

Schwarz, G. E. (1978). Estimating the dimension of a model. *Annals of Statistics, 6(2)*, 461–464.

Steele, R. J., & Raftery, A. E. (2009). *Performance of Bayesian Model Selection Criteria for Gaussian Mixture Models.* Washington, DC: Dept. Stat., Univ. Washington, Tech. Rep. 559.