Universiteit Leiden
Opleiding Informatica

Analyzing flight recorder data:

A data-driven safety analysis of mixed fleet flying

Name:	Laurens Jansma
Date:	08/06/2016
1st supervisor:	dr. M. van Leeuwen
2nd supervisor:	prof. dr. A. Plaat
Assessor:	A. Dijkstra MSc

BACHELOR THESIS

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Analysing flight recorder data:
A data-driven safety analysis of mixed fleet flying

L. D. Jansma

June 8, 2016

Abstract

A major airline recently acquired a new airplane: the Boeing 787. To achieve more operational efficiency, this plane is flown by pilots already flying the Boeing 777. However, the airline wants to make sure that this practice does not influence flight safety. This is done by analyzing the landings, which led to the following research question: does mixed fleet flying of Boeing 777 and Boeing 787 airplanes influence landing performance on Boeing 777 airplanes??

Previous research on machine learning and flight recorder data focused almost exclusively on detecting anomalies. We use machine learning techniques on Boeing 777 flight recorder data to determine if there is a difference in performance between mixed fleet flying pilots and regular pilots, more specifically in the landing phase of the flight. We used both features proposed by experts and automatically constructed features.

Although our techniques were able to distinguish the two subtypes of Boeing 777 airplanes as a proof of concept, a substantial difference in pilot performance was not found in this data set using the techniques presented in this research. These findings support the idea that mixed fleet flying of Boeing 787 and Boeing 777 airplanes does not impact pilot performance.

Contents

List of Tables

List of Figures

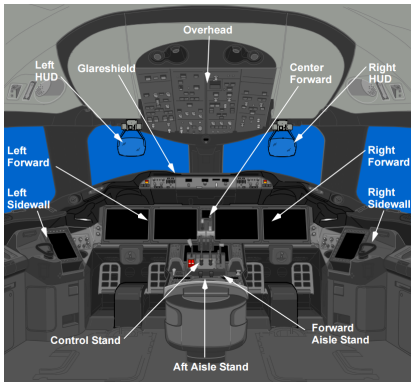
1 Introduction

A major airline has recently acquired a new airplane: the Boeing 787-900 Dreamliner. The new airplane is to be operated by pilots that now fly the Boeing 777-200ER and Boeing 777-300ER airplanes: a somewhat uncommon practice called mixed fleet flying. In this section we first explain mixed fleet flying and the differences between the Boeing 777 and Boeing 787 airplanes in brief, and then discuss the airline's safety practices. We then explain the machine learning techniques to be used in this research, and conclude with the research questions and expected results.

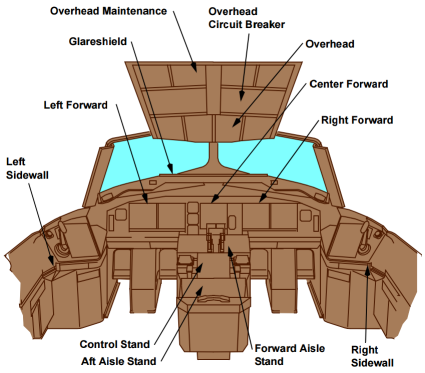
1.1 Mixed fleet flying

The airline decided to let part of the Boeing 777 flight crews operate the Boeing 787 airplanes as well. As of June 2016, approximately 25% of the Boeing 777 pilots are also trained to fly a Boeing 787. Mixed fleet flying gives the airline a big advantage in terms of flexibility: on different days, a different type of plane can be scheduled as needed for the demand. The flight crew planning does not need to change, since pilots can fly either plane.

The main question that arises is: can a pilot that is used to flying a Boeing 777 sometimes fly a Boeing 787 instead, without consequences for their performance? Even though the two types of airplanes have a very high degree of commonality from a flight crews perspective, there are substantial differences [1]. In Figure 1, the differences between some general items in both cockpits are highlighted. These figures are used in the 'differences handout', an internal document provided to mixed fleet flying pilots in training.



(a) Boeing 787 flight deck



(b) Boeing 777 flight deck

Figure 1: A comparison of the Boeing 787 and Boeing 777 flight decks.

In addition to the change in cockpit display panels and the addition of a head-up display, some other differences are noteworthy as well. The Boeing 787 has slightly smaller dimensions (1 meter shorter and 1 meter smaller) than a Boeing 777-200ER, but is way smaller than a Boeing 777-300ER (which is 11 meter longer and 5 meter wider).

Apart from these observable differences, several differences are only noticeable when actually operating the aircraft. For example, while the engine start procedure has to be executed manually in the Boeing 777, the only way to start the Boeing 787's engines is to use an autostart system. Other specific system differences can be listed in a 60-page manual, which is quite comprehensive: for comparison, the entire Boeing 787 FCOM[1] covers, without amendments and additional bulletins, over 1700 pages.

As a result of the high degree of commonality, qualified Boeing 777 pilots can be trained for flying the Boeing 787 in five days (a 'type difference training'). This is possible because, apart from the stated differences, the planes really work in the same way: the two planes have similar cruise speeds, nearly identical flight maneuvers, the same takeoff and landing technique and the same autoland and non-precision approach procedures [3].

[1] Flight Crew Operations Manual

1.2 Flight safety

The airline's mission is to offer reliability with a safe, efficient, service-oriented operation with a proactive focus on sustainability. Safety and efficiency are two parts of the mission that are relevant for this research: efficiency in the Boeing 777 and Boeing 787 operations is partly achieved through flexibility (as explained previously in Section 1.1).

The airline wants to guarantee safety by having an industry-leading risk and performance-based Integrated Safety Management System (ISMS). This is achieved through the identification of hazards, the analysis and mitigation of risks, and promotion of safety awareness and safe behaviour throughout the organisation [2].

This research is part of the analysis of risks: if, despite the commonality of the two planes as discussed in Section 1.1, a substantial difference in landing performance between regular pilots and mixed fleet flying pilots can be detected in the Boeing 777 data, there could be safety risks involved in mixed fleet flying that the airline had not foreseen. In that case, the airline should take measures to reduce the exposure to these risks (risk mitigation).

The biggest part of a flight is done in autopilot mode. This can be considered safe and, moreover, a difference between mixed fleet flying pilots and regular pilots should not be present when a computer is flying the plane. Domain experts expect to find the biggest differences between the groups of pilots in the landing phase. This is why this research will focus on analyzing the landing phase of the flights, more specifically from 200 feet above the ground until the taxiing phase begins.

1.3 Machine learning: decision trees

As we explain in further detail in Section 3, it is not feasible to let domain experts analyze the data manually due to the nature of the data (millions of different measurements). Instead, we opt to use machine learning techniques to determine whether there is a noticeable difference between the landing performance of the two types of pilots or not. The way we use machine learning is outlined in more detail in Section 4.

The field of machine learning consists of many techniques, that typically fall in one of two subcategories:

- Supervised learning: based on labeled examples, the algorithm induces rules to predict the class of newly (unlabeled) presented examples.
- Unsupervised learning: examples are not labeled, and the algorithm has to group the examples into, for example, clusters

For this research, we use a specific technique in the supervised learning category: a decision tree. We choose to use decision trees for two major reasons: first, they can handle large data sets well and second, they generate an easy to interpret model, unlike other classifiers such as Support Vector Machines or neural networks that operate more like a black box.

The generation of an easy to interpret model is a result of the way a decision tree algorithm builds a classifier. Algorithms that construct decision trees usually work top-down, by choosing a variable at each step that best splits the set of items [18]. The best split is usually determined by the information gain, based on the concept of entropy from information theory. The information gain of a node is calculated by substracting the weighted sum of the entropy of the children from the parent's entropy.

1.4 Research questions

As described in Section 1.2, the main objective of this research is to determine whether or not the practice of flying two types of airplanes leads to noticeably different landings. To guide this research, we defined the following main research question:

Research question 1. *Does mixed fleet flying of Boeing 777 and Boeing 787 airplanes influence landing performance on Boeing 777 airplanes?*

To answer this first research question we formulated two other, more specific, research questions:

Research question 2. *Is it possible to distinguish two different types of pilots, by using machine learning techniques on complex and heterogeneous data sets such as Boeing 777 flight recorder landing data?*

Research question 3. *If distinguishing two different types of pilots using this information is possible, which features are the most informative?*

1.5 Expected results

In the best scenario for the airline, no substantial differences are detectable between the two types of pilots. This means that low-scoring evaluation measures are favorable, since this would mean that no substantial differences are present. We expect this to be the case, since the airplanes' manufacturer (Boeing) advertises the mixed fleet flying possibilities in its magazine [3]. In addition, if major negative influences were experienced during simulator training, the airline probably would not have continued the mixed fleet flying project.
However, if the machine learning algorithms would be performing quite well (i.e., high evaluation measures), this would mean a difference actually is present and the airline might want to reconsider its mixed fleet flying operations.

1.6 Structure of this thesis

In the remainder of this thesis, we first discuss related work in the field of applying machine learning techniques to flight recorder data in Section 2. We then describe the data we used in this research and how the data set was prepared for use in Section 3. Next, we provide an outline and in-depth explanation of our approach in Section 4. This is followed by discussing the results of the experiments in Section 5. We then discuss the results and the implications for the airline in Section 6. We finish by answering the research questions and explaining the contributions to flight safety, and discuss further research that can be done in Section 7.

2 Related work

Only limited scientific research has been done on the specific subject of this thesis, namely applying machine learning to flight recorder data. The research that has been done, was usually performed on relatively small data sets and had a different goal: anomaly or outlier detection.

In a paper on aviation data mining, Pagels uses machine learning techniques on a data set partly generated in a simulator to find patterns and anomalies "that indicate potential incidents before they happen" [16]. This purpose of Pagels' research differs from our thesis, as explained further at the end of this section. In addition, for Pagels' research a simulator was used to get data on what a 'bad' landing, with flaps up instead of down for example, would look like. In this thesis, only real-life flight recorder data is used. No noteworthy incidents occurred on the flights in our data set, apart from very few hard landings.

Other research on anomaly detection done by Das et al. was based on a big real-life flight recorder data set (> 25000 landings). This study shows that features based on symbolic dynamic filtering (SDF) can discover anomalies in flight recorder data [6]. In addition, Li et al. used cluster analysis to detect anomalies in flight recorder data [11].

However, this thesis does not concern finding anomalies in flight data: we want to classify two groups of pilots (mixed fleet flying or 'regular'), based on their way of landing a Boeing 777. As far as we know, this is the first attempt to automatically detect differences in landing behavior between two groups of pilots.

3 Data

The data set we obtained from the airline contains about 800,000 rows and 35 columns. In this chapter, we explain how that data is collected and what steps are needed to make the data suitable for use. This preprocessing is done using the python module `pandas`[2]. Section 3.1 explains how the data is collected from airplanes. In Section 3.2, a description of the data is provided. Section 3.3 explains the steps needed to prepare the data for further analysis and how weather data was added to the data set.

3.1 Collection of airplane data

When an airplane is in operation, a lot of values (such as altitude, heading, speed and other mandatory parameters[4]) are recorded and saved to both the Flight Data Recorder (FDR) and an optical disk (Quick Access Recorder). When the airplane returns to the airline's main airport, a ground engineer takes the optical disk out of the airplane and stores it along with other QARs. These are transported to flight operations personnel on a daily basis. They copy all data to the airline's databases. Some modern types of aircraft are able to transfer the data via a wireless or cellular (3G / 4G) connection. After transfer, a processing algorithm compares the values found in the new data to preset values that are considered normal and if certain values exceed the normal values ('exceedance filtering'), one or more experts can decide to analyze what has caused the irregularity. Data of flights that do not show irregularities are stored as well and can be retrieved when needed.

3.2 Description of the airplane data

A data set of approximately 9,000 airplane landings was provided to us by a major airline. The landings were made between July 28, 2015 and March 23, 2016 at 39 different airports all over the world. We used the measurements made during each landing of all fifteen Boeing 777-200ER and all eleven Boeing 777-300ER airplanes operated by the airline, while the altitude above ground was below 200 feet and the speed was above 30 knots. This is the last phase of the landing procedure. We use the measurements made until the point where the airplane begins taxiing.

[2] `http://pypi.python.org/pypi/pandas/`

For the purposes of this research, these variables can be divided into three different categories:

(a) Those that tell something about the pilot's way of landing (e.g., touchdown distance, rate of descent, G-forces)

(b) Those that do influence how the plane lands, but do not tell something about the pilot's performance (e.g., airplane weight, flaps)

(c) Those that do not influence landings directly (e.g., date)

The table in Appendix 1 describes the different column names, their data types and brief descriptions of the measurement.

3.3 Augmenting weather data

Every 30 minutes, weather stations at an airport generate a report describing the local weather conditions. These weather reports are known as METARs: **Met**eorological **A**erodrome **R**eports. Typical METAR data contains information about (for example) the temperature, wind speed, visibility, air pressure and visibility. This data is used by pilots in their weather briefings.

A data set containing all METAR data for each flight's destination at the time of each landing was readily available from the airline. While constructing this data set, the half-hour difference between two reports was accounted for. If, for example, a plane lands at 12:15, the METARs from 12:00 and 12:30 are averaged.

For use in this research, only the wind speed was found to be useful. Other information like the visibility was only available for one-third of the landings, and for only fifteen of the 9,000 landings, snow or ice was present on the runway. We think this is not informative enough to include in the data set. Measurements like the temperature and the barometric pressure do not influence landing performance directly, so we decided not to include this data either. As a result, only the wind speed was augmented to the data set.

4 Approach and methods

In this chapter, we discuss the methods that we use for the experiments. To perform the experiments, we use the Python modules `scikit-learn`[3] [17] and `inspyred`[4]. We first provide a brief outline of the approach. Then, we discuss using aggregate values and the difficulties of working with imbalanced data. We conclude with an explanation regarding the two methods used to determine what features to use and how the analysis was performed.

4.1 Aggregate values instead of time series

For the classification algorithm we intend to use (a decision tree), we cannot use the 'raw' time series as they are available in the data set since decision trees do not support time series. Decision tree algorithms use one attribute per node to make a split in the data set, which cannot be done on time series by default. To solve this problem, we apply some standard aggregate value operators like maximum, minimum, average, etc. on all airplane parameters that the pilot can influence directly via manual input. These are (1) Control Column input, (2) Roll, (3) Pitch, (4) Vertical Speed, (5) Rudder Pedal input and (6) Vertical acceleration.

This approach results in one row of features per landing, with a landing ID as 'index column' followed by several aggregate values. We combine six columns with seven operators, yielding 42 aggregate values per landing. Applying a set of operators to existing features is called feature construction [13]. It goes without saying that it is faster to calculate all combinations of features and operators once and then test which combination performs best, rather than calculating a set of aggregate values over and over. As will be described in Section 4.3, five domain experts first selected the features they expected to be usable for identifying two types of pilots. In Section 4.5, we explore a more automated approach for selecting features.

4.2 Working with imbalanced data sets

Since only a relatively small number of pilots (about 1 in 4) working at the airline is a mixed fleet flying pilot, the number of landings by mixed fleet flying pilots and regular pilots is quite different. Approximately 6500 landings were done by regular pilots, while approximately 2500 landings were

[3] `http://scikit-learn.github.io/stable`
[4] `https://pypi.python.org/pypi/inspyred`

done by mixed fleet flying pilots. This imbalance can cause problems: since classification algorithms tend to minimize error rates, most examples will be classified as belonging to the majority group. This is comparable to a biased roulette wheel: if 75% of the pockets on the wheel would be red, a gambler will most likely bet on the ball falling into one of the red pockets, rather than a black pocket.

However, this does not mean learning from imbalanced data sets is impossible: some studies show that classifiers trained on imbalanced data sets perform the same as other classifiers trained on a sample of the same data set [5], [10]. In their paper on imbalanced data sets, He and Garcia discussed multiple methods to work with imbalanced data [9]. Out of the discussed methods by He and Garcia, we choose to use the undersampling method. This means that some of the landings done by regular pilots are not used for training the classifier.

To apply the undersampling method, we first determine the amount of landings done by mixed fleet flying pilots, and then randomly sample the same amount of landings from the landings done by regular pilots. Each time the algorithm is executed, a different sample of landings made by non-mixed fleet flying pilots is used. The resulting data set is shuffled. As a result, a data set consisting of ± 2500 landings done by regular pilots and ± 2500 landings done by mixed fleet flying pilots is used for training classifiers. Since all landings done by mixed-fleet flying pilots are already used for inducing the classifiers, no samples remain for testing the classifier. In order to avoid over-fitting, we apply 10-fold cross validation.

4.3 Features based on experts' knowledge

To determine which features are most suitable for classifying pilots according to the experts, we interview five persons working at the airline's Flight Safety department. They are a mixed fleet flying pilot, a Boeing 747 pilot, a data scientist, a flight data engineer and a safety investigator. They are presented the columns available as seen in Appendix 1 and are asked what features they thought were suitable to detect differences in landing performance. The features that are named at least two times, and the results when using these features are further discussed in Section 5.3.1.

4.4 Outline of automatic feature construction

To run a classification algorithm, we need to select some features that are used to build a classifier. We use two ways of obtaining these features: interviewing experts (described earlier) and automatic feature construction. This latter method is considerably more complex, and thus we provide a brief overview before explaining this method in detail in Section 4.5.

The outline of this approach can be explained as follows:

1. Calculate the required aggregate values
2. Select all (N) landings done by mixed fleet flying pilots and select N landings done by non-mixed fleet flying pilots
3. Run a genetic algorithm that determines the best settings for the feature selection genetic algorithm

   Step 1. Select and mutate a list of features

   Step 2. Evaluate performance of feature selection genetic algorithm

   Step 3. Tweak settings of the feature selection genetic algorithm

   Step 4. Return to 'Step 1' unless fitness does not improve anymore
4. Return best set of features

4.5 Feature construction & genetic algorithms

In addition to the features proposed by domain experts, we want to test the performance of other combinations of features as well. However, testing all combinations of features would take too much time. Given the 42 aggregate values per landing, and two choices per value (to use or not to use), there is a total of 2^{42}, or approximately 4.4 trillion, combinations. It goes without saying that exploring each combination of features is not feasible.

Since we already calculated each possible combination of values and operators as described in Section 4.1, we now need to find the best combination of features. Earlier research outlines the approach for this 'constructive induction' as three components working together: (1) a machine learning algorithm, (2) a constructive induction module, and (3) an evaluator [8]. Mierswa and Morik introduced the application of genetic algorithms for feature construction [15]. In this research, we combine the classic approach

of combining values and operators with genetic algorithms. We use genetic algorithms to determine the best subset of features. A short explanation of genetic algorithms is provided below.

A genetic algorithm mimics the way evolution works in nature. More fit individuals in a population have a higher chance of surviving and reproduction, while less fit individuals tend to die earlier and, as a result, they will become extinct at some point. Each generation, some individuals are selected to form a new generation. This selection is based on the fitness of the individuals.

The fitness of a certain combination of features is calculated by a so-called fitness function, which is designed by the programmer. After selecting the parents, new offspring is generated using crossover and mutation methods. This cycle of selection and reproduction continues until a stopping point, or termination criterion, is reached. Individuals are expressed as a bit string of 42 bits: one bit for each possible feature as constructed in Section 4.1. An example bit string is: `[0, 0, 1, ..., 0, 1]`. A `0` on index N means the Nth feature is not used, while a `1` would mean the feature is used.

The ways in which individuals are selected, crossed over and mutated are not fixed. The available options for selection, crossover and mutation and the parameters for each method (e.g., population size and crossover probability) are all customizable. To be certain that the selected options for the genetic algorithm yield the best results and the choices made by the programmer do not pose a bottleneck, we use a recipe from the `inspyred` library[5] that determines the best options and parameters: this is called meta-evolutionary computation [7]. This is done by trying each combination of settings (e.g., different variators) and varying the parameters (e.g., mutation rate, population size) each run until the fitness does not increase anymore.

This influences the computation time greatly, since the genetic algorithm is not executed once, but up to tens of thousands of times. In addition, each time the genetic algorithm to determine the features is executed the classification algorithm is ran thousands of times as well. A standard desktop computer needed just over 60 hours to complete 100,000 evaluations.
In short, a meta-evolutionary computation program determines the best selector, operators, parameters, etc., instead of having a human being doing so.

[5] `http://pythonhosted.org/inspyred/recipes.html`

The evaluation function used in this meta-evolutionary algorithm determines what features the specified individual has selected (denoted by a 1 in the bit string) and runs a decision tree classifier using the list of selected features. The accuracy of the classifier determines the biggest part of the fitness but, since we want to use as little features as possible to avoid the "curse of dimensionality", a penalty of 0.005 percentage points is applied for each used feature.

After the meta-evolutionary computing algorithm is finished, both the individual yielding the best fitness and the settings/parameters combination used to generate this individual are returned. The proof-of-concept for this approach is discussed in Section 5.2 and actual results of this approach in identifying mixed-fleet flying pilots are outlined in Section 5.3.2.

4.6 Combining feature construction & expert knowledge

After evaluating both the automatically constructed features and those proposed by domain experts, we combined both lists of features in order to find which features were the most informative. To determine this, we used three lists of features: (1) the list of features chosen by domain experts, (2) the features selected by the genetic algorithm and (3) a combination of list 1 and 2 (without duplicates). We first determined the accuracy of a decision tree using all features (baseline) and then reran the classification algorithm with one feature left out. The difference in accuracy was calculated, and the feature giving the least loss was removed. This was done recursively, removing one feature in each iteration. The results of this approach are discussed in Section 5.3.3.

5 Experiments

In this chapter, we discuss the outcome of the experiments as described in Section 4. First, we explain the performance measures that are used to determine the quality of a classifier. Next, we compare the performance of features chosen by experts with the features that were automatically constructed. We then combine these features and determine the most informative features, i.e. the features that influence the accuracy the most. We conclude this chapter with a month-by-month comparison of evaluation measures, to make sure there was no difference in the first months of flying the Boeing 787 either, since this possible difference might have disappeared later on.

5.1 Evaluation measures

To compare the two approaches explained in the previous chapter, we use five evaluation measures: accuracy, precision, recall, F-measure and the area under the ROC curve (AUC). We briefly explain these features in this section. These values are derived from the confusion matrix. The confusion matrix (Figure 2) consists of the following values:

- TP: True Positive (pilot correctly identified as 777 & 787-pilot)
- TN: True Negative (pilot correctly identified as only 777-pilot)
- FP: False Positive (777-pilot identified as flying 777 & 787)
- FN: False Negative (777 & 787-pilot identified as only flying 777)

	Predicted Mixed Fleet	**Predicted Regular**
Actual Mixed Fleet	True Positive (TP)	False Negative (FN)
Actual Regular	False Positive (FP)	True Negative (TN)

Figure 2: A Confusion Matrix

With these values, we can compute at least five measures that give an indication of the quality of the model. We use the following measurements to compare the different combinations of features (those chosen by experts and those that were automatically constructed).

- Accuracy $= \dfrac{TP + TN}{TP + TN + FP + FN}$

  Quantifies the total percentage of correct classifications.

- Precision $= \dfrac{TP}{TP + FP}$

  Indicates the proportion of pilots that are mixed-fleet flying pilots, among all pilots who were classified as 'mixed-fleet flying pilot'.

- Recall $= \dfrac{TP}{TP + FN}$

  Indicates the proportion of all mixed-fleet flying pilots that were classified as 'mixed-fleet flying pilot'.

- F-measure $= 2 * \dfrac{precision * recall}{precision + recall}$

  The F-measure is the harmonic mean of the precision and recall.

- AUC: The ROC plots the true positive rate against the false positive rate. The area under the curve is a measurement for the usefulness of the classifier: a score of 1.0 means a classifier can be considered perfect, while scores near 0.5 mean the classifier performs as bad as flipping a coin.

5.2 Airplane classification

To make sure our proposed method works, we first try to classify the type of airplane. As explained in the introduction, the airline that supplied the data has two types of Boeing 777 airplanes: the Boeing 777-200ER and the Boeing 777-300ER. The Boeing 777-300ER is 34 feet (10 meters) longer than the Boeing 777-200ER and it weighs 119.000 pounds (54.000 kg) more. We expect that this difference in airplane characteristics is visible in the landing data.

The amount of landings made with both types of planes is more balanced than the amount of landings by mixed / non-mixed pilots, so under-sampling data is not needed for this experiment (there were 5300 landings with a Boeing 777-200ER, and 3600 landings with a Boeing 777-300ER).

The meta-evolutionary computing algorithm returned the following parameters for determining the best features to classify an airplane type:

Parameter	Value
Population size	64
Selector	Default selection
Replacer	Steady state replacement
Crossover	N-point crossover
Mutator	Gaussian mutation
Crossover rate	0.62
# crossover points	4
Gaussian st. dev.	0.96
Mutation rate	0.03

Table 1: Genetic algorithm parameters for airplane type classification

With these parameters, the following features are determined to be best for classifying the type of airplane:

- Maximum control input
- Minimum control input
- Sum of the control input
- Minimum vertical speed
- Standard deviation of the vertical speed
- Sum of the vertical speed
- Highest absolute value of the vertical speed
- Minimum pitch angle
- Medium pitch angle
- Minimum vertical acceleration
- Average vertical acceleration
- Sum of the vertical acceleration
- Maximum roll
- Standard deviation of the roll

The classifier (decision tree) using these features returns the following evaluation measures:

Accuracy	88.15%	
	Score class '772'	Score class '773'
Precision	0.903	0.852
Recall	0.893	0.866
F-measure	0.898	0.859
AUC	0.889	0.889

Table 2: Evaluation measures for airplane type classification

5.3 Pilot classification

As mentioned before, the goal of this research is to determine whether or not mixed fleet flying pilots and 'regular' pilots are distinguishable by applying machine learning techniques to data from the Flight Data Recorder. To do so, we first asked five experts what features they assumed to be useful to classify the two types of pilots. After building a model using these features, we also try an automatic feature construction algorithm as explained in Section 4.5.

5.3.1 Based on domain knowledge

As explained in Section 4.3, we interviewed five experts to determine what features they think are the most promising for classifying two kinds of pilots. The following features were chosen by more than two experts:

- Maximum pitch angle
- Minimum pitch angle
- Maximum roll
- Maximum vertical speed
- Maximum rudder
- Maximum vertical acceleration

Another proposed feature is the so-called 'flare altitude'. When a plane is very close to touchdown, the pilot pulls the nose up: this is called flaring.

In a Boeing 777, this should be about 60 feet above the runway. However, since the Boeing 777 airplanes tend to approach the runway in a slightly steeper angle, pilots have to flare earlier than in a Boeing 787: this results in a higher flare altitude. In a Boeing 787 a pilot can flare a bit later, which results in a lower flare altitude. It is suspected that when pilots have flown in a Boeing 787 as well, their flare altitude can become slightly lower than non-mixed fleet flying pilots.

We tested the performance of the decision tree classification algorithm on the proposed list of features, including the flare altitude. The returned evaluation measures are shown in Table 3:

Accuracy	55.68%	
	Score regular	Score MFF
Precision	0.554	0.560
Recall	0.581	0.532
F-measure	0.567	0.546
AUC	0.534	0.556

Table 3: Evaluation measures for pilot classification, based on domain knowledge

5.3.2 Based on automatically constructed features

The meta-evolutionary computing algorithm returned the following parameters for determining the best features to identify a mixed fleet flying pilot:

Parameter	Value
Population size	5
Selector	Tournament selection
Replacer	Default replacement
Crossover	Heuristic crossover
Mutator	Gaussian mutation
# Selected	5
Tournament size	5
Gaussian st. dev.	0.27
Crossover rate	0.71
Mutation rate	0.31

Table 4: Genetic algorithm parameters for pilot classification

The feature set with the highest fitness consisted of the following features:

- Max. control column input
- Sum of vertical speed
- Median vertical speed
- Maximum pitch
- Sum of pitch
- Median pitch
- Highest absolute value of pitch
- Maximum vertical acceleration
- Minimum vertical acceleration
- Average vertical acceleration
- Median vertical acceleration
- Minimum roll

Using these features for our classifier, the evaluation measures were as follows:

Accuracy	58.12%	
	Score regular	Score MFF
Precision	0.554	0.663
Recall	0.832	0.331
F-measure	0.665	0.441
AUC	0.601	0.601

Table 5: Evaluation measures for pilot classification using automatic feature construction

5.3.3 Combining features

Now that we have found two sets of features, we want to combine them and find the most informative features as described in Section 4.6. The results of finding the best features for automatic construction are described in Table 6, the results for finding the best features of those proposed by the experts are described in Table 7, and the results of the best features of the combined feature sets are described in Table 8. The Δ value shows the increase or decrease in accuracy caused by removing this feature from the data set, with respect to the previous set of features. For the first line, this is the difference between the feature set with the first item removed versus the baseline (the complete feature set).

[6]Percentage points

Baseline	57.11%	
Least-loss features	New accuracy	Δ w.r.t. previous set
1. Pitch sum	57.37%	+0.26 p.p.[6]
2. Roll min	57.64%	+0.27 p.p.
3. Pitch highest absolute	57.61%	−0.03 p.p.
4. Pitch max	57.63%	+0.02 p.p.
5. Vert. acc. median	58.10%	+0.47 p.p.
6. Vert. acc. max	57.50%	−0.60 p.p.
7. Vert. acc. min	56.89%	−0.61 p.p.
8. Control column max	58.42%	+1.53 p.p.
9. Pitch median	57.10%	−1.32 p.p.
10. Vert. acc. average	57.97%	+0.87 p.p.
11. Vert. speed median	61.79%	+3.82 p.p.

Table 6: Best features selected from the features proposed by automatic feature construction

Baseline	54.63%	
Least-loss features	New accuracy	Δ w.r.t. previous set
1. Pitch min	54.28%	−0.35 p.p.
2. Vert. speed max	54.50%	+0.22 p.p.
3. Flare altitude	55.71%	+1.21 p.p.
4. Roll max	54.86%	−0.85 p.p.
5. Vert. acc. max	56.07%	+1.21 p.p.
6. Rudder pedal max	60.58%	+4.51 p.p.

Table 7: Best features selected from the features proposed by experts

Baseline	57.17%	
Least-loss features	New accuracy	Δ w.r.t. previous set
1. Vert. speed max	57.44%	+0.27 p.p.
2. Pitch max	57.88%	+0.44 p.p.
3. Vert. acc. min	57.23%	−0.65 p.p.
4. Vert. acc. median	57.51%	+0.28 p.p.
5. Pitch min	58.16%	+0.65 p.p.
6. Pitch highest absolute	57.71%	−0.45 p.p.
7. Pitch sum	57.15%	−0.56 p.p.
8. Flare altitude	57.69%	+0.64 p.p.
9. Vert. acc. average	57.64%	−0.05 p.p.
10. Rudder pedal max	57.41%	−0.27 p.p.
11. Roll min	57.44%	+0.03 p.p.
12. Vert. speed median	56.82%	−0.62 p.p.
13. Vert. acc. max	57.06%	+0.24 p.p.
14. Roll max	56.60%	−0.46 p.p.
15. Control column max	58.75%	+2.15 p.p.
16. Pitch median	61.79%	+3.04 p.p.

Table 8: Best features selected from the combination of both feature sets

Month	Accuracy	Δ average	σ
July	55.81	-0.65	−1.29
August	56.82	+0.36	+0.72
September	56.07	-0.39	−0.77
October	57.24	+0.78	+1.55
November	56.46	0	0
December	57.05	+0.59	+1.17
January	56.45	-0.01	−0.03
February	56.57	+0.10	+0.21
March	55.68	-0.78	−1.55

Table 9: Accuracies, difference with the average and the amount of standard deviations from the average.

5.3.4 Classification per month

Boeing 787 simulator training began in July, and the airline started operating the Boeing 787 in November. Since our data set spans the period between July and March, there might be a possibility that due to habituation, a difference that was present between the two groups of pilots in the first few months, disappeared later on. To check this, we ran the classification algorithm on a data set for each month separately.

If the classifier accuracy is higher, a greater difference between the two groups of pilots is present since the classifier's distinguishing abilities have improved. For each month, a new set of features to use was determined using automatic feature construction. The resulting average accuracy was 56.46% and the standard deviation of the accuracies was 0.506. Further results are described in Table 9 and Table 10.

In addition to the algorithm evaluation measures, we observed the features that were selected by the genetic algorithm. The most-used (used in more than half of the months) features and the months they were used for are listed in Table 10. A double line is placed between October and November to indicate the start of actual operation of the Boeing 787 in November.

Feature	Jul	Aug	Sep	Oct	Nov	Dec	Jan	Feb	Mar
Pitch Max			X	X	X		X	X	X
Vert. Acc. Abs[7]	X		X	X	X	X			X
Roll Abs					X	X	X	X	X
Roll Med[8]		X			X	X		X	X
Vert. Acc. Sum		X	X			X		X	X
Pitch Sum	X	X		X	X				X

Table 10: The most-used features and the months they were used in

[7] Highest absolute value
[8] Median value

6 Discussion

In this section, we discuss what the outcomes of the classification algorithms imply for the mixed fleet flying project. We first briefly discuss the results of the experiments, and then provide an interpretation by domain experts on the set of automatically constructed features.

6.1 Analysis of the results

While our method is proven to be successful for classifying the type of airplane as shown by the relatively high evaluation measures presented in Section 5.2, it is not able to classify the two groups of pilots accurately. As shown in Table 3 and Table 5, most evaluation measures have values between 0.50 and 0.60. Scores in that range are not nearly enough to say there is a substantial difference between the two groups of pilots: it is only slightly better than a coin flip.

To compare the results in more detail, we present the scores of both classification methods (using features based on expert knowledge and automatically constructed features) next to each other in Table 11. The best scores for each class (regular or mixed fleet) are displayed in bold.

Feature set	**Domain knowledge**		**Automatic construction**	
Accuracy	55.68%		58.12%	
	Score regular	Score MFF	Score regular	Score MFF
Precision	0.554	0.560	0.554	**0.663**
Recall	0.581	**0.532**	**0.832**	0.331
F-measure	0.567	**0.546**	**0.665**	0.441
AUC	0.534	0.556	**0.601**	**0.601**

Table 11: Side-by-side comparison of the evaluation measures for pilot classification.

It is worth mentioning that for detecting mixed fleet flying pilots, the features proposed by the domain experts have a higher precision and recall than the automatically constructed features. For the classification of regular pilots, however, the automatically constructed features had a better recall than the features proposed by the experts.

The high **precision** of the automatically constructed features means that it returned more relevant results than irrelevant results when classifying mixed fleet flying pilots. For classifying regular pilots both feature sets yielded the exact same precision, so the amount of relevant cases was the same.

When comparing the values for **recall**, the opposite is true. The automatically constructed feature set has a much higher score for returning regular pilots than for mixed fleet flying pilots. This means that when classifying regular pilots, most of the relevant cases are returned while this was not the case when classifying mixed fleet flying pilots.

The differences between precision and recall are higher in the automatic feature construction set than in the domain knowledge feature set. The **F-measure** is a combination of precision and recall, used to judge the trade-off between them. The high F-measure for classifying regular pilots with the automatically constructed features shows that a classifier built with this set of features does not miss a lot of the regular pilots. The automatically constructed feature set does, however, miss a lot of mixed fleet flying pilots. Since the difference in F-measures is smaller in the feature set proposed by the domain experts, the trade-off between precision and recall is more balanced there.

The last measure used is the **AUC**. The area under the curve is 0.601 for both types of pilots when using the automatically constructed features. This result can be regarded as poor (≥ 0.60), while the even lower AUC for the feature set based on domain knowledge can be regarded as a 'fail' (< 0.60).

6.2 Most informative features

Despite the not very strong performance of the decision tree algorithm using our features, it is good to know the features that influence the classification of pilots the most: these can be seen as the most informative features for classifying the type of pilot. Knowing this allows the airline to let instruc-

tors pay specific attention to these details when training new mixed fleet flying pilots. Suppose the airplane's roll is a very distinctive factor due to high roll values in mixed fleet flying pilots' landing performance. When instructors know that pilots tend to let the airplane roll more when landing a Boeing 777, they can incorporate a warning for this behavior into the type difference training.

As described in Section 4.6, we combine both lists of features and test which features are influence the accuracy of the decision tree the most. The results of this experiment are shown in Table 8. Two features that influence performance the most, and thus are removed last, are the maximum control column input and the median pitch angle. An expert's opinion (mixed fleet flying pilot) on these observations is discussed in Section 6.4.

A noteworthy result of removing features is that removing the last feature in the list leads to a big increase in accuracy (+3 to +4.5 p.p.). We are not completely sure what causes this increase, but we expect that this is a result of the greedy approach of the decision tree algorithm. Initially, the algorithm can make a lot of splits on lots of variables and does so in a greedy way (the local optimum). When fewer features are available, this might lead to the algorithm splitting on other features, yielding higher accuracy.

6.3 Per-month classification results

In Table 9, we described the results when splitting the data set into one data set for each month and then running a decision tree algorithm on the data set. In Table 10, the features selected by the genetic algorithm for each month's data set were presented.

A notable observation is that for the months July to October, when the Boeing 787 was not yet in operation, the genetic algorithm seldom chose to use the highest absolute and median 'Roll' values. However, as soon as the Boeing 787 came into operation, the highest absolute value of the roll became a feature that was selected in each month, while the median roll value was only left out once.

We suppose that the sudden usage of this feature is due to the fact that the Boeing 787 rolls slightly less than the Boeing 777 when turning the control column the same amount. An expert interpretation on this is provided in the next section.

As for the evaluation measures as described in Table 9, classification accuracy differed a lot between months. Major differences with the average score were visible in October ($+1.55\sigma$) and March (-1.55σ). However, we cannot explain what has caused these differences. Similar chosen features and score differences were returned when running the per-month analysis once more.

6.4 Interpretation by a domain expert

We asked a mixed fleet flying pilot (Captain) what could make the roll values apparently more standing out after the beginning of Boeing 787 operations. Since the pilot has experience on both planes (3,000 flying hours on the Boeing 777 and 11,000 in total), he can be considered an experienced pilot.

The pilot explained that there is a difference in the fly-by-wire system (control laws) between both planes. While a Boeing 777 reacts to a pilot steering left and then immediately right by moving the airplane left and immediately right (leading to a more or less shaking airplane), the Boeing 787 fly-by-wire system neutralizes such quick inputs.

The pilot expects pilots to use the control column more when landing a Boeing 777 than when landing a Boeing 787. Fewer control input should lead to less roll. If this hypothesis is true, the algorithm might be able to detect a *non*-mixed fleet flying pilot based on higher absolute roll values. Unfortunately, we cannot test whether pilots use the control column on a Boeing 787 more than on a Boeing 777 or not, since we only have a data set containing Boeing 777 landing data.

The apparent difference in maximum pitch could not be explained by the pilot based on the control laws.

7 Conclusions

Now that we have described the results of the experiments and discussed these in detail, we can answer the research question. By answering the research questions, we have made some contributions to flight safety: both specifically for the airline's mixed fleet flying project and more in general on detecting differences in pilot behavior based on flight recorder data. Of course, research on this subject is not yet finished, thus we conclude by discussing possible future work.

7.1 Answers to the research questions

For clarity, we repeat the original research questions as described in Section 1.4, immediately followed by an answer to the question based on our experiments, results and discussion.

Research question 1. *Does flying a Boeing 787 (in addition to a Boeing 777) influence landing performance when flying a Boeing 777?*

> The influence of flying a Boeing 787 on Boeing 777 flying (more specifically: landing) performance is, apparently, very limited. Our classification algorithm could hardly distinguish the two types of pilots, with scores below 0.60 or below 60%. As such, we can say that based on this data and experiments, flying a Boeing 787 in addition to a Boeing 777 does not influence landing performance on a Boeing 777.

Research question 2. *Is it possible to distinguish two different types of pilots, by using machine learning techniques on complex and heterogeneous data sets such as flight recorder data from landings with a Boeing 777?*

> To a certain extent, machine learning techniques (in this case, a decision tree) are able to distinguish two types of pilots. The evaluation measures, however, were not convincing. Since some pilots were correctly classified as being a mixed fleet flying pilot, we can still say that it is possible to distinguish two types of pilots, just not very accurately.

Research question 3. *If distinguishing two different types of pilots using this information is possible, which features are the most informative?*

> Based on the results as presented Table 8 and Table 10 and further discussed in Section 6.4, we can conclude that the most informative features are the maximum pitch, the highest absolute roll and the maximum vertical acceleration.

7.2 Contributions to flight safety

As for the general contributions to flight safety, this is the first time – to our knowledge – that flight recorder data is used to detect differences between two groups of pilots. The same methods can potentially be used to identify the effect of, for example, additional trainings. The only thing that needs to be done, is to label each pilots' landing with a 0 or a 1, depending on whether or not the pilot has participated in the training. The algorithm will classify each pilot, and if the effect of the training is high enough to be detectable (high evaluation measures), one can say that the training has altered pilot behavior substantially.

More specifically for the airline's safety mission, we have shown that (based on the provided data and used methods) the effect of mixed fleet flying on landing a Boeing 777 is negligible. As such, no additional measures should have to be taken, at least when looking at the data set and experiments used in this research.

Due to the nature of the deliverables to the airline, the experiments can be repeated using newer data sets each month. When the algorithm evaluation measures suddenly rise, there might be a growing difference in mixed fleet flying pilots' performance.

7.3 Future work

In this research, we used aggregate values to build decision trees. In further research, the complete time series can be used to detect differences between pilots. There are multiple ways to do this, one of which is to use the Symbolic Aggregate approXimation (SAX) method as proposed by Lin and Keogh [12]. Other ways for applying machine learning to time series data are available as well, of course.

The parameters for our decision tree classifier were the default parameters as selected by `scikit-learn`. In future work, these parameters can be optimized to achieve higher scoring classifiers. Other future work might concern using completely different classification or clustering algorithms, like Support Vector Machines or Multiple Kernel Anomaly Detection [6]. Although we do not know the performance of our decision tree classifier when compared with other methods, a comparison might be interesting.

In addition, data that includes the 'on final approach' phase can be used as well. Some pilots start flying manually from approximately six miles from the runway (usually on an altitude of $\pm$2000 ft), and behavior in this phase might prove to be useful as well.

Another way in which mixed fleet flying could influence a pilot's performance, is during non-regular situations. The effect of mixed fleet flying on dealing with incidents can be tested in a simulator, by comparing the way a regular pilot handles an incident in a Boeing 777 with a mixed fleet flying pilot handling the same incident. The same approach as presented in this thesis can be used for this.

8 Bibliography

Documents

[1] Airline's 777/787 Flight Technical department: *787-9 and 777-200ER & 300ER FCOM Differences Handout*, unpublished internal document, March 2015.

[2] Airline's Operational Safety & Quality Assurance department: *Integrated Safety Management Manual*, unpublished internal document, April 2016.

[3] Boeing AeroMag QTR_01 (2008). Retrieved on June 1, 2016 from `http://www.boeing.com/commercial/aeromagazine/articles/qtr_1_08/AERO_Q108_article2.pdf`.

[4] ICAO Annex 6, Operation of Aircraft, Vol I, Attachment D and Vol III.

Research / literature

[5] G.E.A.P.A. Batista, R.C. Prati, and M.C. Monard: *A Study of the Behavior of Several Methods for Balancing Machine Learning Training Data*, in ACM SIGKDD Explorations Newsletter, vol. 6, no. 1, pp. 20-29, 2004.

[6] S. Das, S. Sarkar A. Ray, A. Srivastava, and D. L. Simon: *Anomaly Detection in Flight Recorder Data: A Dynamic Data-driven Approach*, 2013 American Control Conference, Washington DC, pp. 2668-2673, 2013.

[7] A. L. Garrett, Inspyred recipies: Meta-Evolutionary Computation (source code). Retrieved in April 2016 from `http://pythonhosted.org/inspyred/recipes.html#meta-evolutionary-computation`.

[8] G. Gómez and E. F. Morales: *Automatic feature construction and a simple rule induction algorithm for skin detection*, in Proceedings of the ICML workshop on Machine Learning in Computer Vision, 2002.

[9] H. He and E. A. Garcia: *Learning from Imbalanced Data*, in IEEE Transactions on Knowledge and Data Engineering, vol. 21, no. 9, pp. 1263-1284, Sept. 2009.

[10] N. Japkowicz and S. Stephen: *The Class Imbalance Problem: A Systematic Study*, in Intelligent Data Analysis, vol. 6, no. 5, pp. 429-449, 2002.

[11] L. Li, M. Gariel, R. J. Hansman, and R. Palacios: *Anomaly detection in onboard-recorded flight data using cluster analysis*, in Digital Avionics Systems Conference (DASC), 2011 IEEE/AIAA 30th, pp. 4A41 4A411, October 2011.

[12] J. Lin, E. Keogh, L. Wei and S. Lonardi: *Experiencing SAX: a novel symbolic representation of time series*, in Data Mining and Knowledge Discovery, vol. 15, p. 107, 2007.

[13] C. J. Matheus and L.A. Rendell: *Constructive Induction On Decision Trees*, in IJCAI, Vol. 89, pp. 645-650, August 1989.

[14] M. Kantardzic: *Data Mining: Concepts, Models, Methods, and Algorithms*. John Wiley & Sons, New York, 2003.

[15] I. Mierswa and K. Morik: *Automatic feature extraction for classifying audio data*, in Machine learning vol. 58, no. 2-3, pp. 127-149, 2005.

[16] D. A. Pagels: *Aviation Data Mining*, in Scholarly Horizons: University of Minnesota, Morris Undergraduate Journal, vol. 2, no. 1, article 3. 2015.

[17] F. Pedregosa et al.: *Scikit-learn: Machine Learning in Python*, in Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011

[18] L. Rokach and O. Maimon: *Top-down induction of decision trees classifiers - a survey*, in IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 35, no. 4, pp. 476-487, November 2005.

9 Appendix 1: Measurements used

As mentioned in Section 3.2, the columns in the data set are explained in this appendix. Each column name is mentioned, along with their data types, units, a brief description and a category. The category letter is used to indicate the effect of the concerned column on landing and means the following:

(a) Those that tell something about the pilot's way of landing (e.g., touch-down distance, rate of descent, G-forces)

(b) Those that do influence how the plane lands, but do not tell something about the pilot's performance (e.g., airplane weight, flaps)

(c) Those that do not influence landings directly (e.g., date)

The only column that is not labeled as explained above is the 'Mixed fleet flying pilot' column: the purpose of this research was to determine whether mixed fleet flying affects landing performance. Since this was not known in advance, we labeled this column with a question mark.

Column name	Data type	Unit	Category	Brief description
Day	Integer	-	c	Day of the month
Month	Integer	-	c	Month number
Tail	String	-	c	Unique airplane identifier
Departure airport	String	-	c	4-letter ICAO code of the departure airport
Destination airport	String	-	c	4-letter ICAO code of the destination airport
Mixed fleet flying pilot	Boolean	-	?	Indicates a pilot also flies Boeing 787s
Runway	String	-	c	Runway used for landing
Radio altitude	Integer	Feet	a	Altitude above ground
Pressure altitude	Integer	Feet	a	Altitude above sea level
Pitch	Float	Degrees	a	Vertical nose position
Heading	Float	Degrees	a	Airplane direction relative to the North
Roll	Float	Degrees	a	Horizontal position of the airplane
CCW	Float	Degrees	a	Control column input (no input = 0.0)
Rudder Pedal	Float	-	a	Rudder pedal usage
Rudder	Float	Degrees	a	Position of the rudder on the airplanes tail
Thrust left and right	Float	Degrees	a	The position of the throttle levers
N1	Float	Percent	a	The amount of thrust produced (% of maximum)
Reversers	Boolean	-	a	Indicates deployment of the reversers
V_ref	Integer (ordinal)	Knots	b	Calculated optimal landing speed
Indicated airspeed	Float	Knots	a	Speed including wind effects
Groundspeed	Integer	Knots	a	Speed relative to the ground
Distance	Integer (ordinal)	Feet	a	Touchdown distance from the beginning of the runway
Vertical speed	Integer	Feet per minute	a	Descent speed
Stabilizer	Float	Degrees	a	Vertical stabilizer setting
Autopilot	Boolean	-	b	Indicates autopilot status (active / not active)
Vertical acceleration	Float	g	a	G-force measured
Flaps	Integer (ordinal)	Degrees	a	Flaps setting (usually 30)
Weight	Float	Tonnes	a	Calculated airplane weight
Temperature	Float	Celcius	a	Outside temperature
Nose Gear	Boolean	-	a	Indicates if the nose gear position (air / ground)
Main Gear	Boolean	-	a	Indicates if the nose main position (air / ground)
AirGround	Boolean	-	a	Indicates airplane position (air / ground)

Table 12: A description of all variables in the data set