Twelfth USA/Europe Air Traffic Management Research and Development Seminar (ATM2017)

A novel machine learning model to predict abnormal Runway Occupancy Times and observe related precursors

Floris Herrema & Vincent Treve & Bruno Desart
Airport research unit
EUROCONTROL
Brussels, Belgium
floris.herrema@eurocontrol.int
vincent.treve@eurocontrol.int bruno.desart@eurocontrol.int

Richard Curran & Dries Visser
Air Transport and Operations
TU Delft
Delft, Netherlands
r.curran@tudelft.nl h.g.visser@tudelft.nl

***Abstract* - Accidents on the runway triggered the development and implementation of mitigation strategies. Therefore, the airline industry is moving toward proactive risk management, which aims to identify and predict risk precursors and to mitigate risks before accidents occur. For certain predictions Machine Learning techniques can be used. Although many studies have explored and applied novel Machine Learning techniques on different aircraft Radar and operational Taxi data, the identification and prediction of abnormal Runway Occupancy Times and the observation of related precursors are not well developed. In our previous papers, three feasible methods were introduced: Lasso, Multi-Layer Perception and Neural Networks to predict the Taxi-Out Time on the taxiway and the Time to Fly and True Airspeed profile on final approach. This paper presents a new Machine Learning method, where we merge these feasible Machine Learning techniques for predicting the abnormal Runway Occupancy Times of unique radar data patterns. Additionally we use in this study the Regression Tree method to observe key related precursors extracted from the top 10 features. Compared with existing methods, the new method no longer requires predefined criteria or domain knowledge. Tests were conducted using runway and final approach aircraft radar data consisting of 78,321 Charles de Gaulle flights and were benchmarked against 500,000 Vienna flights.**

Keywords-component; combined machine learning technique, (abnormal) AROT, regression tree and precursors

I. Introduction

Machine Learning (ML) can be used to identify patterns[1] and to observe related risks precursors in past data [1, 2, 3, 4]. These patterns and precursors can be transferred into 'what-if' statements by analyzing relations between the Arrival Runway Occupancy Time (AROT) [5] and the prediction variables highlighted in Table 1. This analysis is needed to predict forthcoming operational risks during real time landing operations based on the observation of risks precursors. Such a prediction would feed predictive tools at the airport to alert Air Traffic Controllers (ATCO's) about impending aircraft behaviors and to produce both point forecasts and probabilistic forecasts in real time. In this context, a decision-support tool is needed to inform the ground controller risks associated to the AROT. This study will mainly focus on a better understanding and prediction of *abnormal* AROT's which lead to an improvement in runway safety and throughput.

A. *Related work*

Many work in the literature have tried to tackle these patterns and observe related risks precursors. We divide the literature in two groups. We start with the first group which focuses on abnormal events detection methods for aviation systems. The Morning Report software package was one of the earliest efforts made to detect abnormal events from routine Flight Data Recorder (FDR) data [6]. The software models the time-series data of selected flight parameters using a quadratic equation. Each flight is mapped into a point that is described by the coefficients of the quadratic equations in the feature space. Thereafter, for each flight an 'atypical score' is measured using the distance between the point and the mean of the distribution in the feature space. Later studies apply data-mining techniques to detect abnormal events in data of aerospace systems [7–12]. One of these studies applies supervised learning software called; Inductive Monitoring System (IMS) [10]. The IMS software method summarizes the data distributions of typical system behaviors from a historical training dataset, which is then compared with real-time operational data to detect abnormal behaviors. However, the limitation of the IMS is that it always needs a training data set for labeling the norms. Other studies used unsupervised learning techniques. These studies focuses on discrete flight parameters for monitoring pilot operations such as cockpit switch flips [7, 8]. The techniques observe abnormal events in the switch operations based on the longest common subsequence measures. The study [9] developed a statistical framework to incorporate both continuous and discrete flight parameters in FDR data. Das et al. developed multiple kernel abnormal detection (MKAD), which applies a one-class

[1] *Patterns in this context are defined as the normal distributions (from -2 to +2 sigma) of Runway Occupancy Time for a given flight and for a range covering from crossing the threshold and its tail vacating the runway exit.*

support vector machine for abnormal detection [12]. MKAD assumes one type of data pattern for normal operations, which is not always valid in real operations, since standards vary according to flight conditions. Most recently, Matthews et al. summarized the knowledge discovery pipeline for aviation data using the previously discussed algorithms [13]. Also many studies on AROT have been executed such as [14] and [15].

The second group of literature consists of abnormal event detection outside the aviation domain. In general those abnormal event techniques can solve problems for a domain-specific formulation. Some techniques are developed for intrusion detection in computer systems [16-18]; fault detection in mechanical units and structures [19,20]; and fraud detection related to phones, credit cards, insurance claims [21, 22], etc. Additionally, two groups of techniques are developed for time-series data depending on how dissimilarities are measured: data based and model based [23, 24]. The former measures the dissimilarity based on data observations. The dissimilarity is measured by a variety of distance functions such as Euclidean distance [25, 26], dynamic time warping distance [27], probability based distance [28], correlation-based distance [29].

B. *Aim*

With the literature in respect, it is envisaged to progressively move from a statistical approach to a new ML approach for coping with the variability of AROT behaviors. In previous papers that we wrote [3, 4], ML techniques were assessed on their capabilities to produce fast and accurate predictions and to test a number of 'what-if' statements.

The main purpose of this study is twofold. First, to better characterize and predict AROT as a function of operational parameters from historical data. Second to identify and predict abnormal AROT flights with their related risks precursors. The identification and prediction is done using a new data-driven method by merging 3 feasible ML techniques. The key objective of this study is to develop a real time model that forecasts the AROT for different aircraft types and weather conditions using this new data-driven method for Charles de Gaulle (CDG) and Vienna (VIE) airport. This model should offer insight into the predictability of key precursors impacting AROT.

C. *Structure*

The structure of this paper is as follows: First, the data and prediction variables are described in section II. Second the methodology is outlined in section III whilst introducing the AROT behavior and data preparation. Furthermore, the feasible ML techniques are combined to a new ML model which is applied on a regression tree. Thereafter abnormal AROT results are predicted and related precursors are observed. In section IV the real time model is outlined and finally conclusions are drawn in section V.

II. DATA AND PREDICTION VARIABLES

The AROT is a key driver of airport runway throughput, especially when low airborne separation minima are applied. Several factors, such as aircraft type, weather conditions (wind and visibility), traffic demand, air traffic controller workload, and the coordination of flows with neighbouring airports influence the AROT [14, 15]. In order to predict AROT profiles and extract risks precursors, final approach and runway radar data are used.

A. *Final approach and runway radar data*

Aircraft radar data is extracted from runway schedulers and has been provided by CDG and VIE airport. The data covers respectively 5 years and 3 years of final approach and runway radar data from 2011 to 2015 and from 2013 to 2015 included. In total, the data comprises about 78,321 and 500,000 arrival flights. The data sets are stored in CSV formats and are thereafter saved in separate MatLab and Python files.

Table 1. Prediction and response variables.

	AROT Variables	***Description***
Prediction variables	1. Anne	Year
	2. Caractredevol	Commercial or private flight
	3. CodeIATA	IATA code company
	4. CodeAeroportOACI	Airport origin ICAO code
	5. CodeAeroportIATA	Airport origin IATA code
	6. Compagnie	Airline
	7. Crosswind	Crosswind vector
	8. DateReal	Actual date
	9. Deep landing	The runway length available beyond the touchdown point
	10. IdentifiantvolATC	ATC call sign
	11. Long flare	Estimate the start of the flare until touchdown
	12. Mois	Month
	13. NumFlight	Flight Number
	14. Postedestationnement	Gate arrival
	15. QFU	Runway Orientation
	16. Semaine	Week
	17. Tailwind	Tail wind vector
	18. Temp	Temperature
	19. TimeReal	Actual time of the day
	20. Typeavion	Aircraft type
	21. Visibility	METAR visibility conditions
Additional prediction variables	22. Arrival runway throughput	The amount of landings that is performed on a runway in 30 min
Identification of AROT	23. ACSpeedPoint	Speed of the aircraft at 2NM, and 1NM out, threshold and the runway exit

Prediction variables	*AROT Variables*	*Description*
	24. ALDT	Actual Landing Time
Response variable	25. AROT	Arrival Runway Occupancy Time

III. METHODOLOGY

For this study, we propose a methodology comprising five steps. The method is based on our previous papers [3, 4], literature study [30] and the Statistical Package for the Social Science (SPSS) ML method [31]. This methodology describes the steps to come up with a usable predictability model that identifies and predict abnormal AROT flights with their related risks precursors. Each step is detailed below:

A. *Identification and understanding of the AROT*

The AROT response is extracted by calculating the time between the aircraft crossing the threshold and its tail vacating the runway exit [5], using the variables ACSpeedPoint and ALDT. This forms a matrix $\boldsymbol{Y}$ where each row represents a flight, column 1 until 22 a prediction variable and column 25 the AROT response variable (table 1).

We propose to first extract the AROT per aircraft type for runway 09L, 27R, 08R and 26L for all 78,321 CDG flights and for 500,000 VIE flights for runway 11L and 29R. This is done to cover seasonal variations and to have a minimum of 15 AROT measurements per aircraft type and runway. Figure 1 shows for 49 aircraft types their AROT at runway 08R at CDG airport. The AROT results for the remaining CDG runways and 49 different aircraft types can be found in [30].

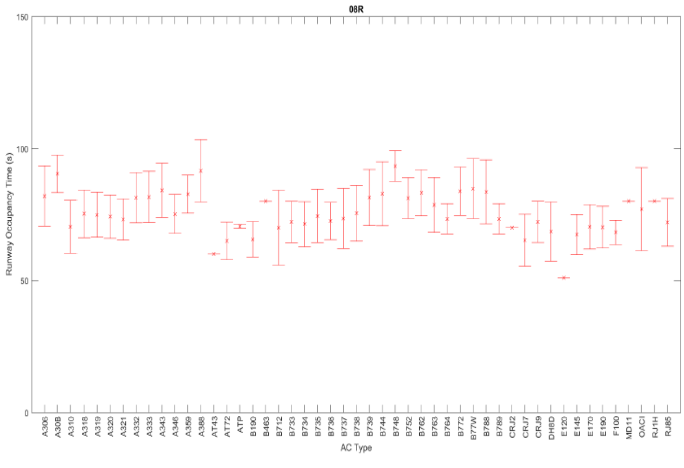


Figure 1; Example of AROT per aircraft type for runway 08R

We observed from Figure 1 that the AROT is influenced by the aircraft type or different aircraft categories. Therefore as a next step we plot the AROT as function of the categories; ‘Heavy’, ‘Medium’ and ‘Small’ per CDG runway and time of the day.

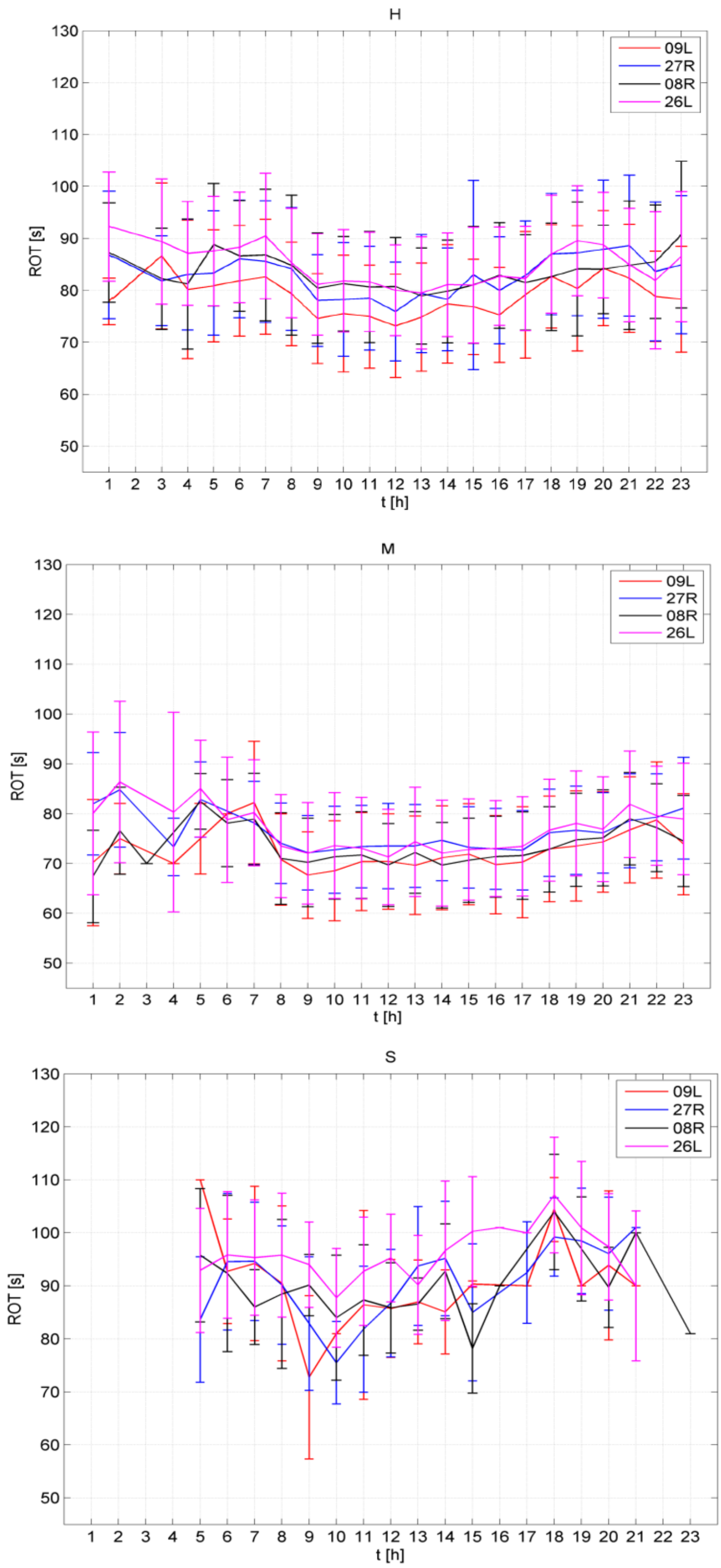


Figure 2; AROT versus time of the day and different ICAO aircraft categories (H, M, S) for runway 09L, 27R, 08R and 26L.

Figure 2 shows that there is an effect of the hour of the day (peak hours) on the extracted AROT. The most likely explanation for this behaviour is the differences in runway throughput or weather conditions (day/night). Plotting the arrival throughput versus the AROT shows that a higher throughput leads to a decrease in the AROT for all categories. Figure 3 shows an example for the 'Medium' aircraft category.

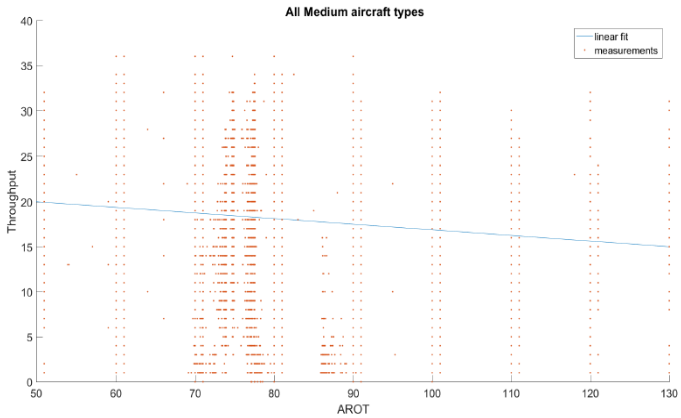
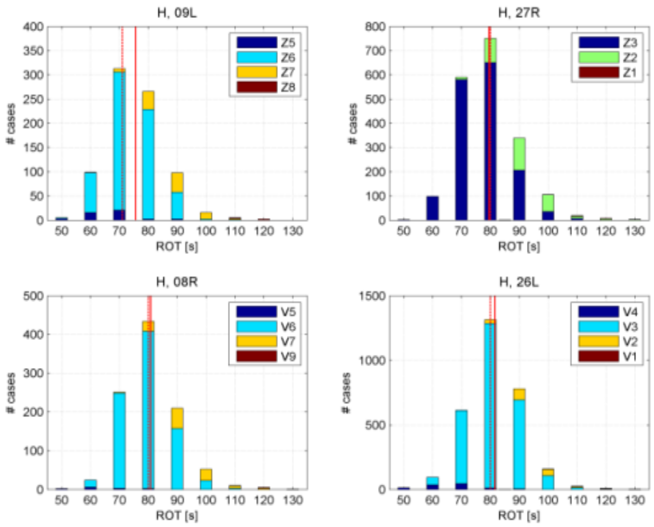


Figure 3; AROT versus different runway throughput levels for different aircraft types on runway 26R.

According to [15] the AROT depends mainly on the approach speed, the Rapid Exit Taxiway (RET) used, the type of RET, the exit speed and the brake policy by the airlines. Therefore, as a next step we plot the AROT versus different CDG runway exits for both ICAO and RECAT-EU (Figure 4). All results can be found in report [30].



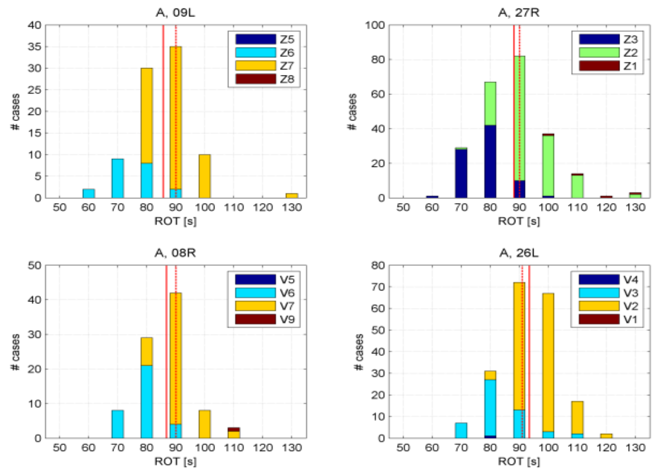


Figure 4; Number of flights versus the AROT for different runway exits for Heavy aircraft (H-ICAO) and type A (RECAT-EU) aircraft.

B. Data preparation

The data preparation phase covers all activities required to set up the final dataset from the initial raw aircraft operational runway and final approach radar data. Before the ML model will be trained with the prediction variables highlighted in Table 1, first the most important (group) features will be selected using RreliefF and Sequentialfs (feature selection). In previous papers we have introduced these techniques [3, 4]. The objective of feature selection is threefold: improving the prediction performance of the predictors, providing faster and more effective predictors, and providing a better understanding of the underlying process that generated the data [32]. RreliefF and Sequentialfs have commonly been viewed as feature selection methods that are applied in a prepossessing step before the model is learned [33]. The standard RreliefF regression modelling technique has been extensively discussed in many papers [34]. In this study, the technique has been applied on 78,321 final approach flights for CDG as shown in Figure 5 and benchmarked against 500,000 Vienna flights. We observed for both airports that the QFU, aircraft type, arrival runway throughput, visibility, wind vectors and temperature are the most important features for predicting the AROT. The same results are obtained for the Sequentialfs technique.

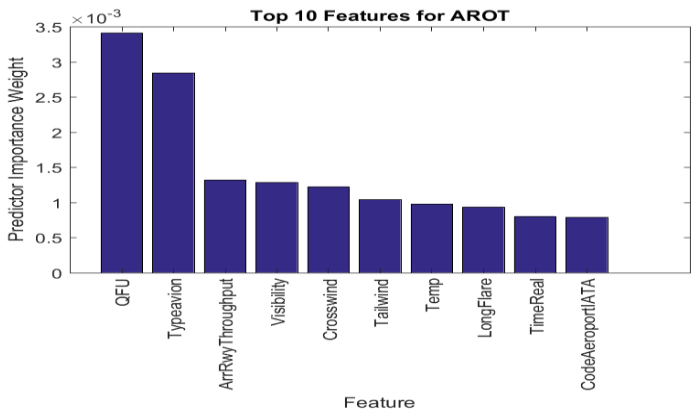


Figure 5; Top10 features for the AROT using the RreliefF technique for CDG.

Thereafter we construct the datasets and find the stability of three different data parts. Based on different data sources and the Table 1 mentioned variables, standardize the feature matrix *X*. Split the matrices *X and Y* in two subsets *Xtrain; Ytrain;* used to train the model and *Xtest;Ytest* used to evaluate the model accuracy. Finally we analyse the default ratios (splitting the data) for training, testing and validating into respectively 70%, 15% and 15%.

C. *Merging feasible ML techniques*

To answer the question 'What machine learning algorithm should I use and how should I combine them?' Depends on the size, quality and nature of the data. It depends what you want to do with the answer. It depends on how the math of the algorithm was translated into instructions for the computer you are using and it depends on how much time you have. Even the most experienced data scientists can't tell which algorithm will perform best before trying them [35].

This study will use the outcome of 3 feasible ML techniques; Lasso, MLP and Neural Networks. By doing so we take into account the characteristics of final approach and runway radar data. The method is based on expert studies [35, 36, 37] and will now be explained in 3 steps;

<u>Novel combined ML method</u>

- First we learn the AROT for 78,321 flights with 22 different features. Three models will be learned using the Lasso, Multi-Layer perception and Neural Networks technique.
- Second we merge the AROT results of all 3 models to one final matrix for 78,321 flights.
- Finally we apply the regression tree technique to the final matrix obtained in the previous step.

D. *Assessing combined ML method*

We have seen in previous papers that learning a model with less prediction variables wouldn't change the MSE significantly compared to with all the variables. Therefore we propose to identify abnormal AROT flights based on the 10 important features [5]. Before we analyze the forecast performance, computational time and minimum amount of data needed for the novel combined ML technique, we first check the stability of three different data parts known as cross validation. To check the stability of different data parts, the data will be randomly divided into training, validation and testing subsets. It has been assumed that the default ratios in this study for training, testing and validation are 0.70, 0.15 and 0.15, respectively. The model is adjusted accordingly during training. The validation is used to measure network generalization and to halt training when generalization stops improving. To prove that a randomly selected data set is stable, epoch and validation checks are performed. Epoch indicates the amount of a single pass through the entire training set, followed by testing of the verification set. Thereafter we check convergence on the validation and at the end of the learning process the model is evaluated on the test set. The test has no effect on the training and therefore provides an independent measure of network performance during and after training. Figure 6 shows an example of a trained model by selecting 78,321 CDG final approach flights for runway 08R. We learned the model with the 10 most important prediction variables highlighted in figure 5. It has also been tested that the same Mean Squared Error (MSE) results are obtained using all features. However by excluding 14 variables the model is trained two times faster.

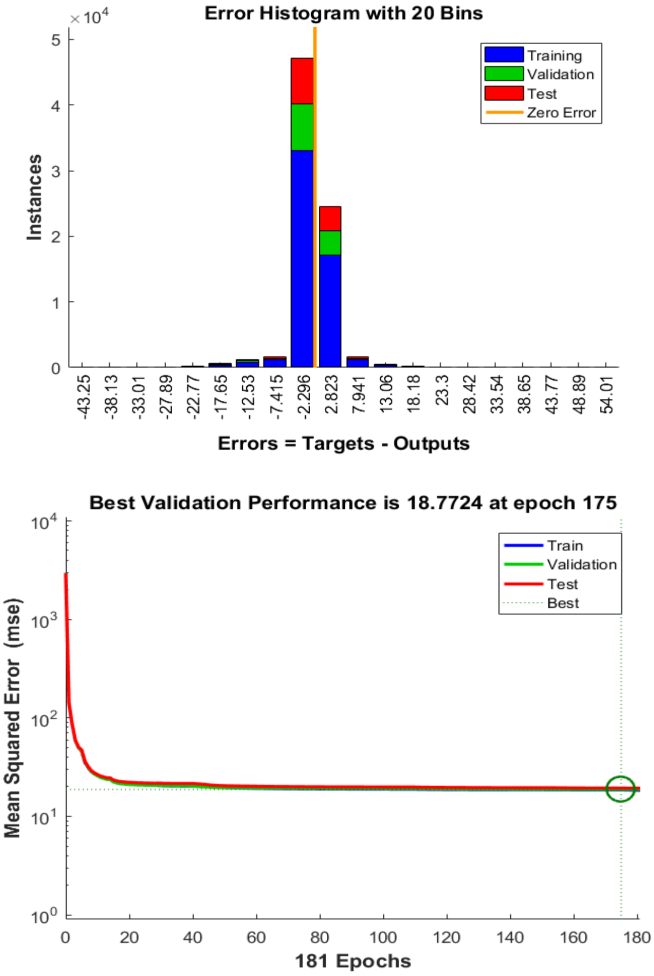


Figure 6; MSE of AROT using the top 10 features.

<u>Abnormal AROT flights and observation of outliers</u>

We are interested in abnormal flights that stay too long on the runway. Therefore, in this study we consider an abnormal AROT if it is +2σ deviated from the normal standard deviation mean. Only those flights are learned with our combined machine learning technique. As a next step, the inconsistency is measured between the outputs and the targets. One way to show this inconsistency is by plotting the regression for the training, validation, test sets and for the complete set (all). Figure 7 shows an example where the regression R values measure the inconsistency between the predicted outputs and the targets. An R value of 1 means a close relationship, 0 a random relationship.

By analyzing this R values we observe outliers. With outliers we mean when a data point is not consistent with the other data points. In this study we assume an outlier when it has an R value between 0 and 0.25. Analysing these graphs shows that there are indeed outliers. It will be obvious that by neglecting them in the target set, a better R value will be obtained for the predicted model. Doing this for the above example results in an overall R value of 0.54 instead of 0.31 presented in Figure 7.

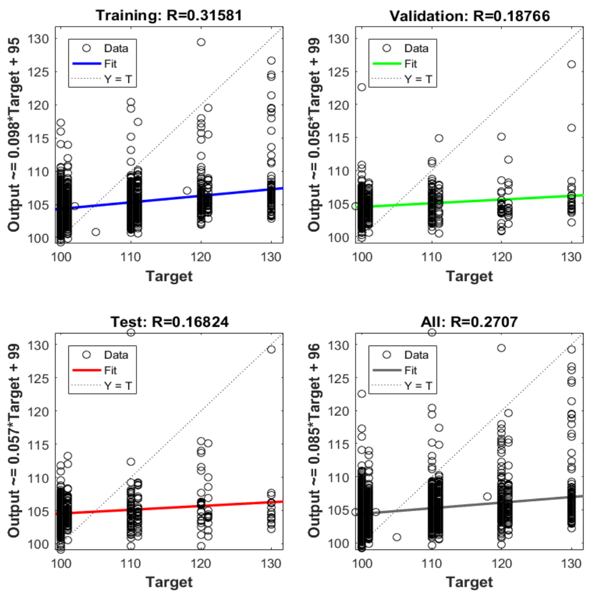


Figure 7; Outliers example of abnormal AROT flights for training the model with 10 features.

The next and final step of our methodology (Step E) will only take those abnormal AROT flights into account with an R value of 0.25<R<1 for runway 08R at CDG.

E. Observe risk precursors with regression tree

The purpose of building a regression tree is to extract a set of if-then-else (what-if statements) split conditions in order to extract the main risk precursors that most influence abnormal AROT flights. The observed flights from figure 7 (0.25<R<1) for runway 08R are analysed and build into a single regression tree which should give a good understanding of which features (top 10 from Figure 5) influence these abnormal AROT flights. By building this tree we start at the root node, and ask a sequence of questions about the predictors. In each iteration, the tree chooses the variable and the split point to achieve the minimum MSE between the predictions and the abnormal AROT targets. This process will continue until a stopping rule is applied. Each of the terminal leaves represents one of the partitions of the input space. To provide a model that can generate accurate predictions and are not over-complicated, we need to find the optimal tuning parameters for the tree. In this study we use two parameters. The first parameter is the *minimum leaf size (l_{min})*, for which we need enough data points in each terminal node to create a distribution. The parameter minimum leaf size can be used to stop the splitting process when the number of instances in a leaf is too small. In addition, if the tree contains too many variables, it is hard to interpret. The second tuning parameter for the tree is the *maximum tree depth (d_{max})*. A very large tree with many leaves might over-fit the data, while a small tree might not be able to capture the important structure of all the variable or top 10 feature variables. The maximum tree depth can restrict the number of layers of a tree. Cross-validation is used to select the minimum leaf size, l_{min} and MSE to select the maximum tree depth, d_{max}. So in our case the tree is fit for a range of values of the two parameters to three quarters data. Thereafter, the MSE of the predictions is computed on the remaining one quarter. This is done for each quarter of the data, and the four MSE are averaged. The set of parameters that gives the lowest MSE will be selected.

As shown in Figure 8, we first train the trees with all 22 variables and different settings of d_{max} and $l_{min.}$ We observed that the MSE drops as the tree depth increases from one to 7, regardless of the leaf size. After tree depth reaches 6, the MSE does not change significantly. On the other hand, a tree with 400 minimum leaf size performs slightly better than the trees with 500 and 600 minimum leaf sizes. We also try to set the minimum leaf size to be less than 400, but the model does not improve much. Moreover, if we further reduce the leaf size, we may not have enough instances in the leaves to fit a distribution. Thereafter we also train a model with the top 10 features shown in Figure 5, where we fit a tree to the entire data set with maximum tree depth and minimum leaf size set to 6 and 400, respectively. We then sort the predictors based on their feature importance and select the first 10 as the final predictors. Thereafter, we retrain the tree with these 10 variables. We change the values of d_{max} and $l_{min.}$ and repeat the cross-validation process described above. The tree with l_{min} equals 400 still performs slightly better than the others, and the MSE does not change significantly after tree depth reaches 6. Thus, our final model, has 10 predictors and is fitted with maximum tree depth and minimum leaf size set to 6 and 400, respectively.

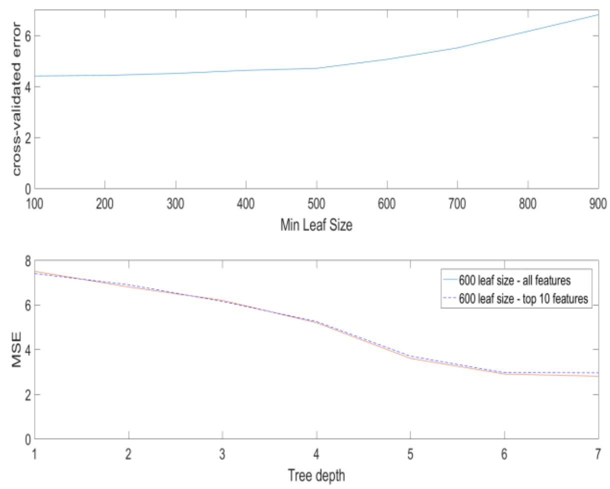


Figure 8; MSE versus Tree Depth for different leaf size and features.

By learning the tree a mean and distribution is extracted per decision node. This is needed to observe risk precursors and understand what is likely to happen for abnormal AROT flights. Our model divides all the abnormal AROT flights into 17 segments. In other words, the regression tree shown in Figure 9 has 17 terminal nodes for which the outcomes are rounded off to 100, 105, 110, 115 or 120 seconds. We can interpret the most important predictors as the major factors that play key roles in influencing abnormal AROT.

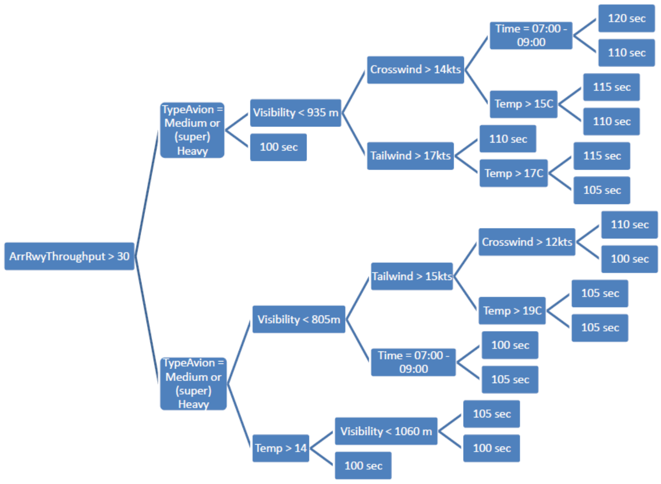


Figure 9; Regression tree only for abnormal AROT at runway 08R. The tree shows 'what-if' statements. If the statement is right we go to the upper node, if the statement is wrong we go to the under node.

After the tree is learned with a tree depth of 6, we observed for the 17 abnormal AROT categories their related precursors which are shown in Table 2.

Table 2. 17 different abnormal AROT categories with their related precursors.

Precursor category	**Likelihood of abnormal flight occurrence per precursor category**	**Amount of flights observed per precursor category**	**Median of the AROT and the Root MSE**
1.ArrRwyThroughput > 30 TypeAvion = Medium or (super) Heavy Visibility < 935m Crosswind > 14kts Time = 07:00 -09:00	9%	33	120 sec 3.5 sec
2.ArrRwyThroughput > 30 TypeAvion = Medium or (super) Heavy Visibility < 935m Crosswind > 14kts Time ≠ 07:00 -09:00	12%	21	110 sec 2.8 sec
3.ArrRwyThroughput > 30 TypeAvion = Medium or (super) Heavy Visibility < 935m Crosswind ≤ 14kts Temp > 15C	3%	18	115 sec 2.9 sec
4.ArrRwyThroughput > 30 TypeAvion = Medium or (super) Heavy Visibility<935m Crosswind ≤ 14kts Temp ≤ 15C	20%	40	110 sec 2.1 sec
5.ArrRwyThroughput > 30 TypeAvion = Medium or (super) Heavy Visibility ≥ 935m Tailwind > 17kts	11%	18	110 sec 3.4 sec
6.ArrRwyThroughput > 30 TypeAvion = Medium or (super) Heavy Visibility ≥ 935m Tailwind ≤ 20kts Temp > 17C	17%	21	115 sec 3.2 sec
7.ArrRwyThroughput > 30 TypeAvion = Medium or (super) Heavy Visibility < 935m Tailwind ≤ 17kts Temp ≤ 17C	15%	18	105 sec 3.9 sec
8.ArrRwyThroughput > 30 TypeAvion ≠ Medium or (super) Heavy	18%	15	100 sec 4.0 sec
9.ArrRwyThroughput ≤ 30 TypeAvion = Medium or (super) Heavy Visibility < 805m Tailwind > 15kts Crosswind > 12kts	21%	23	110 sec 2.2 sec
10.ArrRwyThroughput≤30 TypeAvion = Medium or (super) Heavy Visibility < 805m Tailwind > 15kts Crosswind ≤ 12kts	33%	30	100 sec 2.5 sec
11.ArrRwyThroughput≤30 TypeAvion = Medium or (super) Heavy Visibility < 805m Tailwind ≤ 15kts Temp > 19C	8%	26	105 sec 3.1 sec
12.ArrRwyThroughput≤30 TypeAvion = Medium or (super) Heavy Visibility < 805m Tailwind ≤ 15kts Temp ≤ 19C	16%	11	105 sec 3.2 sec
13.ArrRwyThroughput≤30 TypeAvion = Medium or (super) Heavy Visibility ≥ 805m Time = 07:00 -09:00	20%	20	100 sec 3.5 sec
14.ArrRwyThroughput≤30 TypeAvion = Medium or (super) Heavy Visibility ≥ 805m Time ≠ 07:00 -09:00	9%	15	105 sec 4.2 sec
15.ArrRwyThroughput≤30 TypeAvion ≠ Medium or (super) Heavy Temp > 14C Visibility < 1060m	8%	10	105 sec 2.4 sec
16.ArrRwyThroughput≤30 TypeAvion ≠ Medium or (super) Heavy Temp > 14C Visibility ≥ 1060m	23%	9	100 sec 3.4 sec
17.ArrRwyThroughput≤30 TypeAvion ≠ Medium or (super) Heavy Temp ≤ 14C	19%	20	100 sec 3.2 sec

Given the regression tree in Figure 9, we fit a parametric distribution to each terminal leaf. The probability distributions we considered include the Gumbel, Gamma, and F distributions. The following equation shows the Gumbel distribution since this one fits best.

$$f(x) = \frac{1}{\beta} e^{-\left(\frac{x-\mu}{\beta} + e^{-\frac{x-\mu}{\beta}}\right)}$$

For $-\infty<x<\infty$ where $0 < \mu$, $\text{ß} < \infty$

Figure 10 shows the 17 Gumbel distributions fitted to the terminal leaves. The shapes of the terminal leaves' distributions are quite different from each other. In general, the distributions with lower medians are less spread out. This indicates that in these segments, the uncertainties of the AROT flights are low. If there are a lot of AROT flights in these segments at the airport, the managers should have more confidence in making adjustments to their plans.

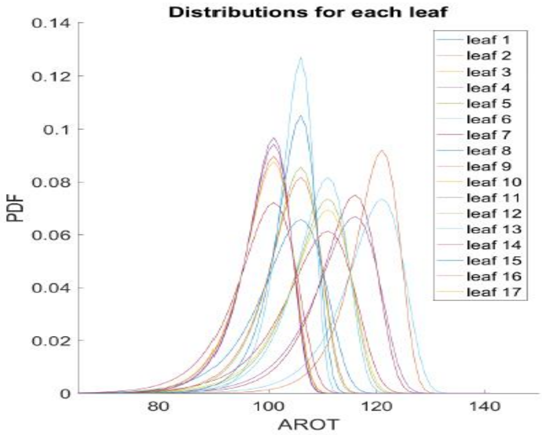


Figure 10; Distributions of the Instances Fall into the 17 Terminal Leafs.

IV. Real Time Model

Based on the data availability we develop a prototype model using the regression tree method to forecast AROT at CDG airport. This model is built on 78,321 flights collected over 5 years Radar data. To generate real-time predictions from the model, we develop an application using MatLab. The output from running the application included the mean and quantiles of flights. The aim is to generate forecasts for each AROT flight and number of landing aircraft to a given runway per time window. Suppose we are at time *h*, and try to make predictions for the next *x* minutes. Given a real-time flight information before ALDT, our regression tree model will determine which segment the flight belongs to. For example, if a small aircraft type plans to land at runway 08R with an ArrRwyThroughput>30, then this flight will fall into leaf 8 of the abnormal AROT flights. Thus, the median of the AROT is 100sec and the distribution of her connection time can be described by a Gumbel distribution with μ =103 and ß=4.0. However there is a likelihood of 18% that it will fall into the abnormal AROT flight category.

Next, we produce the distribution of the number of AROT aircraft during a certain time interval $[h_1,h_2]$ where $h< h_1< h_2<h+x$. This distribution is obtained by aggregating all the distributions of the flights that landed in the last 2 hours or will land in the next *x* minutes. The procedure of generating this distribution is summarized in two steps;

I. Suppose there are *n* flights that landed in the last 2 hours or will land in the next *x* minutes. We sample one AROT from each of the *n* flights' distributions, and calculate the time when the tail is vacating the runway. We then count how many flights landed between the time interval $[h_1,h_2]$, and record this number as y_1.

II. Repeat step I, *m* times, and construct an empirical distribution using $y_1, y_2, \ldots\ldots y_m$. Then the *q-th* quantile of the number of flights land between the time interval $[h_1,h_2]$ can be approximated by the *q-th* quantile of $y_1, y_2, \ldots\ldots y_m$

In the live trail we produced the distributions of the AROT of the flights who have landed in the last 2 hours or will land in the next *x* minutes.

Live trail - assumed Real time data

We assume real time data for a selected data set for which we exclude the actual data variables ALDT ACSpeedPoint. To conveniently generate predictions in real time, we develop a MatLab compiler that can work in most operation systems (Windows, Linux, Mac, etc.). Figure 11 shows the interface of the application.

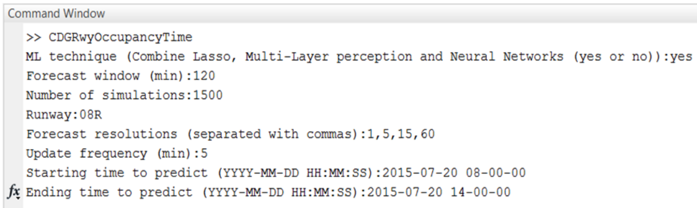


Figure 11. Interface of application.

This application allows users to set the forecasting window *x*, the runway at CDG (RWY), number of simulations (*m*), update frequency, forecast resolution (*r*), starting time of the first forecasting window (*h*), machine learning technique (ML) and ending time of the last forecasting window. The default settings of the first three parameters are 120 min, 1500 simulations, and 5 min, respectively. We update the predictions every 10 min and the default resolutions are 1 min, 5 min, 15min, and 60 min. The starting time defaults to the current time if the user does not specify one. The ending time will be 24 hours after the starting time. As shown in Figure 12, the predictions for this case study are generated on a rolling basis to runway 08R. Suppose the trial started at 8:00 a.m. We first collected data of the flights who landed at CDG after 6:00 a.m. or will land before 10:00 a.m., and then generate forecasts for the next two hours (8:00 – 10.00 a.m.). Thirty minutes later (8:30 a.m.), the second trial started. Similarly, we only considered the flights who landed at CDG after 6:30 a.m. or will arrive before 10:30 a.m., and generate forecasts for the time interval between 8:30 to 10:30 a.m. The dots in Figure 12 show the difference between the predicted AROT and their real values (error). The difference is measured in seconds and shown for 12 flights, for which each flight fall into one of the ICAO categories 'Small' (S), 'Medium' (M) or 'Heavy' (H). We didn't show all the flights in this example but only three per 30 minutes to illustrate the differences per ICAO category.

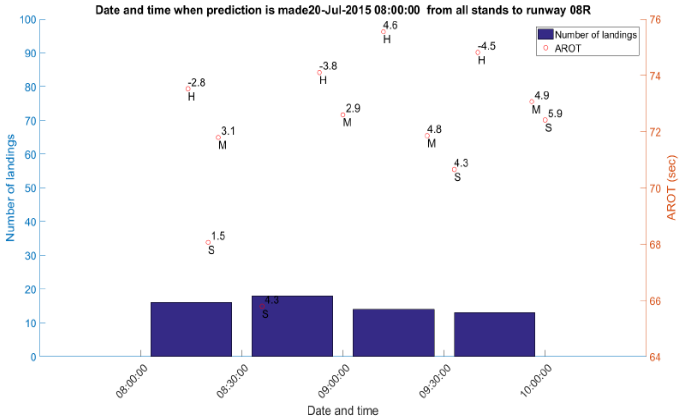


Figure 12. Output after running the application for the first trial.

It has been observed that for this case study and for each prediction trial the first 30 minutes have a significantly lower error compared to the remaining 90 minutes prediction time. Table 4 shows these differences in percentages for four trials.

Table 4. Average error differences per trial and for the time prediction window 0 - 30 minutes and 30 - 90 minutes.

	0 - 30 minutes	**30 - 90 minutes**
Trial 1	6%	10%
Trial 2	4%	8%
Trial 3	4%	9%
Trial 4	2%	6%

V. DISCUSSION AND CONCLUSION

This study demonstrates the use of combined machine learning techniques to forecast AROT per flight. We first reviewed the AROT aircraft behaviours, the data sources, key features and ML outcome. Based on the availability of data and the importance of the problem, we then identified activities that will benefit from making greater use of data, where we focused on the AROT at runway 08R.

We built a predictive model for AROT flights by merging the Neural Network, MLP, Lasso and Regression Tree technique. We then developed an approach to generate distributions of each AROT flight and the number of landings to a specific runway (08R) within a time frame of 30 minutes. We also developed an application for CDG and VIE to produce these forecasts. Finally, we will run a live trial at CDG on the 11th of March 2017, for which we have to assess the accuracy of the model and making improvements. For this we will write a feasibility study where we analyse how our predictability tool can be used by air traffic controllers in their decision making and planning to ensure resilience, safety, and efficiency of air traffic control operations locally in a sector, but also taking into account coordination with other sectors.

After assessing our feature selection techniques RrelliefF and Sequentialfs we include the following 10 features in our ML model; QFU, typeavion, ArrRwyThroughput, Visibility, Crosswind, Tailwind, Temp, LongFlare, TimeReal and CodeAeroportIATA. Based on ATC experience, it is well known that these features are mostly impacting the AROT behaviour. From our regression tree we have learned that by knowing the top 7 features in advance a good prediction can be made of the abnormal and normal AROT for which each abnormal AROT flight fall into one of the 17 precursor categories shown in Table 2. Furthermore, we observed that the regression technique performs best for finding related precursors, for which the regression technique is fitted with a maximum tree depth and minimum leaf size of 6 and 400, respectively.

There are some advantages associated to using our model. First, the machine learning technique we used to build the model is fast, intuitive and efficient. It can help the managers to understand the driving features of the AROT per runway. Second, our model has been built based on a large historical data set of 78,321 CDG and 500,000 VIE flights. For which 22 variables are available for selection as predictors. These variables also enable one to build new features using domain knowledge of the data. Third, our model can update the predictions in real time. The application we developed for CDG and VIE can easily extract real-time data for both airports. The forecasting procedure is effective and the predictions can be generated in a short amount of time. Our model is the first to provide forecasts for each AROT flight to a specific runway. The forecasts of an AROT flight may help ATCO's to make better decisions, predict whether the flights will experience abnormal AROT and to anticipate in advance on the aircraft sequencing on final approach by knowing when landing queuing starts. If an ATCO can retrieve this information far in advance, he or she may be able to generate more stable and accurate AROT used in A-CDM and APOC.

While the model is developed for an AROT problem, we believe the methodology proposed in this study can be easily applied to other runway processes, such as the prediction of unstable approaches. This will be done in our next paper.

Acknowledgment

This work is supported by the EUROCONTROL Experimental Centre and the SESAR programme funded by EU, EUROCONTROL and the aviation industry. The authors would like to thank CDG and VIE that provided access to the aircraft final approach and runway radar data and machine learning expert Massimiliano Zanin for his valuable insights. The author also wishes to acknowledge the support and expertise of the technical teams that made this work possible.

References

[1] Logan, T. J., "Error Prevention as Developed in Airlines," International Journal of Radiation Oncology, Biology, Physics, Vol. 71, No. 1, Jan. 2008, pp. S178–S181. doi:10.1016/j.ijrobp.2007.09.040

[2] Lishuai Li, Santanu Das, R. John Hansman, Rafael Palacios, Ashok N. Srivastava. (2015). Analysis of Flight Data Using Clustering Techniques. *Journal of Aerospace information systems*. 12 (9), p587-598.

[3] F.F. Herrema, V. Treve, R. Curran, H.G. Visser, (2016). Evaluation of feasible machine learning techniques for predicting the time to fly and aircraft speed profile on final approach. *International Conference on Research in Air Transportation*. 8 (4), p4-8.

[4] Floris Herrema, Richard Curran, Hendrikus Visser, Alexei Sharpanskykh, Denis Huet, Bruno Desart, Regis Lacote. (2016). Taxi-Out Time prediction model through machine learning techniques at Charles de Gaulle airport. *SESAR Innovation Days Delft*. 8 (8), 8.

[5] EUROCONTROL. (2012). *Arrival Runway Occupancy Time*. Available: http://www.eurocontrol.int/lexicon/lexicon/en/index.php/Arrival_runway_occupancy_time. Last accessed 30th Sep 2016.

[6] Amidan, B. G., and Ferryman,T. A., "Atypical Event andTypical Pattern Detection within Complex Systems," 2005 IEEE Aerospace Conference, IEEE Publ., Piscataway, NJ, March 2005, pp. 3620–3631. doi:10.1109/AERO.2005.1559667.

[7] Budalakoti, S., Srivastava, A., and Otey, M., "Anomaly Detection and Diagnosis Algorithms for Discrete Symbol Sequences with Applications to Airline Safety," IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews, Vol. 39, No. 1, 2009, pp. 101–113. doi:10.1109/TSMCC.2008.2007248

[8] Budalakoti, S., Srivastava, A. N., and Akella, R., "Discovering Atypical Flights in Sequences of Discrete Flight Parameters," 2006 IEEE Aerospace Conference, IEEE Publ., Piscataway, NJ, March 2006, pp. 1–8. doi:10.1109/AERO.2006.1656109

[9] Srivastava, A. N., "Discovering System Health Anomalies Using Data Mining Techniques," Proceedings of the 2005 Joint Army Navy NASA Air Force Conference on Propulsion, Charleston, SC, June 2005, pp. 1–11, https://c3.nasa.gov/dashlink/static/media/publication/Data_Mining_for_ISHM.pdf.

[10] Iverson, D. L., "Inductive System Health Monitoring," Proceedings of the 2004 International Conference on Artificial Intelligence (IC-AI04), CSREA Press, June 2004.

[11] Schwabacher, M., Oza, N., and Mattews, B., "Unsupervised Anomaly Detection for Liquid-Fueled Rocket Propulsion Health Monitoring," Journal of Aerospace Computing, Information, and Communication, Vol. 6, No. 7, 2009, pp. 464–482. doi:10.2514/1.42783

[12] Das, S., Matthews, B. L., Srivastava,A. N., and Oza, N. C., "MultipleKernel Learning for Heterogeneous Anomaly Detection: Algorithm andAviation Safety Case Study," Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM,Washington, D.C., 2010, pp. 47–56.

[13] Matthews, B. L., Das, S., Bhaduri, K., Das, K., Martin, R., and Oza, N., "Discovering Anomalous Aviation Safety Events Using Scalable Data Mining. Algorithms," Journal of Aerospace Information Systems, Vol. 10, No. 10, 2013, pp. 467–475. doi:10.2514/1.I010080.

[14] Vivek Kumar, Lance Sherry, Rafal Kicinger. (2009). Runway Occupancy Time Extraction and Analysis Using Surface Track Data. -. 10 (3), p3-5.

[15] D.D. Lee, A. Smith, R. Cassell, B. Abdul-Baki. (1999). NASA low visibility landing and surface operations (LVLASO) runway occupancy time (ROT) analysis. -. 8 (2), 25-30.

[16] Hofmeyr, S. A., Forrest, S., and Somayaji, A., "Intrusion Detection Using Sequences of System Calls," Journal of Computer Security, Vol. 6, No. 3, 1998, pp. 151–180. doi:10.1.1.43.6197

[17] Lee,W., and Stolfo, S., "Data Mining Approaches for Intrusion Detection," Proceedings of the 7th USENIX Security Symposium, USENIX Assoc., Berkeley, CA, Jan. 1998.

[18] Mukherjee, B., Heberlein, L. T., and Levitt, K. N., "Network Intrusion Detection," IEEE Network, Vol. 8, No. 3, 1994, pp. 26–41. doi:10.1109/65.283931.

[19] Bishop, C. M., "Novelty Detection and Neural Network Validation," IEE Proceedings of Vision, Image and Signal Processing, Vol. 141, No. 4, 1994, pp. 217–222. doi:10.1049/ip-vis:19941330-vis:19941330

[20] Worden, K., Manson, G., and Fieller, N. R. J., "Damage Detection Using Outlier Analysis," Journal of Sound and Vibration, Vol. 229, No. 3, 2000, pp. 647–667. doi:10.1006/jsvi.1999.2514.

[21] Brause, R., Langsdorf, T., and Hepp, M., "Neural Data Mining for Credit Card Fraud Detection," Proceedings of the 11th IEEE International Conference on Tools with Artificial Intelligence, IEEE Publ., Piscataway, NJ, 1999, pp. 103–106. doi:10.1109/TAI.1999.809773.

[22] Bolton, R. J., and Hand, D. J., "Statistical Fraud Detection: A Review," Statistical Science, Vol. 17, No. 3, 2002, pp. 235–255. doi:10.1214/ss/1042727940

[23] Liao, T. W., "Clustering of Time Series Data—A Survey," Pattern Recognition, Vol. 38, No. 11, Nov. 2005, pp. 1857–1874. doi:10.1016/j.patcog.2005.01.025

[24] Gaber, M. M., Zaslavsky, A., and Krishnaswamy, S., "Mining Data Streams: A Review," ACM Sigmod Record, Vol. 34, No. 2, 2005, pp. 18–26. doi:10.1145/1083784

[25] Gariel, M., Srivastava, A. N., and Feron, E., "Trajectory Clustering and an Application to Airspace Monitoring," IEEE Transactions on Intelligent Transportation Systems, Vol. 12, No. 4, 2011, pp. 1511–1524. doi:10.1109/TITS.2011.2160628

[26] Chan, P. K., and Mahoney, M.V., "Modeling Multiple Time Series for Anomaly Detection," Fifth IEEE International Conference on Data Mining (ICDM'05), IEEE Computer Soc., Houston, TX, 2005, pp. 90–97. doi:10.1109/ICDM.2005.101

[27] Keogh, E., Lonardi, S., and Ratanamahatana, C. A., "Towards Parameter-Free Data Mining," Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, 2004, pp. 206–215. doi:10.1145/1014052.1014077

[28] Kumar, M., andWoo, J., "Clustering Seasonality Patterns in the Presence of Errors," Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, 2002, pp. 557–563. doi:10.1145/775047.775129

[29] Möller-Levet, C., Klawonn, F., Cho, K.-H., and Wolkenhauer, O., "Fuzzy Clustering of Short Time-Series and Unevenly Distributed Sampling Points," Advances in Intelligent Data Analysis V, Vol. 2810, Springer, Berlin, 2003, pp. 330–340. doi:10.1007/978-3-540-45231-7-3-540-45231-7_31

[30] F.F. Herrema (2015). *Big data analysis and machine learning at airports to support decision making*. Brussels: p10-50.

[31] P. Chapman (2000). *Step-by-step data mining guide*. https://www.the-modeling-agency.com/crisp-dm.pdf New York: SPSS. P50-76.

[32] Serial metrics. (2013). *Machine Learning*. Available: http://serialmetrics.com/blog/what-is-machine-learning-feature-selection/. Last accessed 12-02-2016.

[33] MARKO ROBNIK S, & KONONENKO, I. (2013). *Theoretical and Empirical Analysis of ReliefF and RReliefF*. Available: http://link.springer.com/article/10.1023%2FA%3A1025667309714. Last accessed 12-02-2016.

[34] MARKO ROBNIK S, & KONONENKO, I.. (2003). An adaptation of Relief for attribute estimation in regression. (3), 9.

[35] Hyndman, R. (2016). *ARIMA models*. Available: https://www.otexts.org/fpp/8. Last accessed 7th Nov 2016.

[36] Arizona state university (2013). *Design and analysis of experiments*. 5th ed. Arizona: Thomson Digital. 449.

[37] Hai Ning. (2016). *Overview diagram of Azure Machine Learning Studio capabilities*. Available: https://azure.microsoft.com/en-us/documentation/articles/machine-learning-studio-overview-diagram/. Last accessed 7th Nov 2016.