Predicting the amount of air traffic demand regulations using machine learning

A. Doutsis

Technische Universiteit Delft

Predicting the amount of air traffic demand regulations using machine learning

by

A. Doutsis

in fullfillment of the degree of Master of Science
at the Delft University of Technology,

Student number:	4364813	
Project duration:	May, 2019 – December, 2019	
Thesis supervisors:	Dr. M. Mitici,	TU Delft, Air Transport Operations
	R. Sanaei,	DLR Institute of Air Transportation Systems

Preface

This document contains the research and findings of my master thesis research project *"Predicting the amount of air traffic demand regulations using machine learning"*. In this new research, machine learning was applied to predict the evolution of Air Traffic Flow Management measures for demand regulation. At the same time, I try to answer the question "How far into the future can I make predictions?", by utilizing the concept of long-range dependence in stochastic processes. The research and writing of this thesis was conducted from May to December 2019.

The project was conceptualized by the ATFM department of the DLR Institute of Air Transportation Systems and the majority of the research has been conducted during my stay in Hamburg at this institute. The research was challenging and at times even discouraging, given the poor performance of the initial predictive models. However, thanks to the support and guidance of my DLR supervisor R. Sanaei and TU Delft supervisor Dr. M. Mitici, i was able to meet the research objectives within time and quality constrains.

I would like to thank R. Sanaei for his insights in ATFM operations and his suggestions throughout the project. I would also like to thank Dr. M. Mitici for her recommendations in development of the predictive model and her constant encouragement throughout the project to improve upon my scientific work. Without a doubt I must express my gratitude to my family that have supported me throughout my academic endeavours. Finally a big thank you goes for all my friends in Delft, Tirana and Hamburg for their willingness to proofread and discuss scientific concepts, for their constant motivation and positive outlook on life and for making my stay in Hamburg amazing.

A. Doutsis
Delft, December 2019

Contents

I

Scientific article

Predicting the amount of air traffic demand regulations using machine learning

Author MSc student: A. Doutsis[a],
TU Delft supervisor: M. Mitici[a], DLR supervisor: R. Sanaei[b]

[a] *Air Transport Operation Department, Faculty of Aerospace Engineering, Delft University of Technology, Delft, The Netherlands*
[b] *Air Transportation Systems, German Aerospace Center (DLR), Hamburg, Germany*

Abstract

Demand for air transportation is expected to continue growing. Within Europe one of the biggest impacts of this traffic growth, is an increase of air travel delay. As it happened during the summer of 2018, where demand from aircraft intending to enter an air sector was not complemented with capacity to safely accommodate it. Incentivised by this event, in this article the problem of predicting a class of measures for demand-capacity balancing, known as Air Traffic Flow and Capacity Management (ATFCM) regulations, is investigated. A Random Forest model was trained on public ATFCM notification messages to predict the amount of ATFCM regulations over different European air sectors for varying prediction horizons. In addition to the predictive model, in this paper a new way to estimate the maximum prediction horizon is proposed. Using the Hurst exponent, the time-scale at which random behaviour is initiated is found. Comparison of the proposed method with the prediction horizon obtained from the largest Lyapunov exponent indicates that the method is a valid technique for estimating the prediction horizon. By extending the prediction horizon of the model, it is found that the proposed method can reasonably estimate the prediction horizon above which prediction accuracy starts to degrade.

Keywords: Air transport, Demand regulations, Tactical phase, Machine learning, Hurst Exponent, Prediction horizon, Long-range dependence

1. Introduction

According to Eurocontrol's Statistics and Forecast Service in the most likely scenario by 2040 an increase of 53% in IFR movements within Europe can be expected [1]. One of the biggest impacts of this increase in air traffic within Europe, is an increase of air travel delay. An example of this increase of delays was experienced during the summer months of 2018 [2], where the average arrival and departure delays almost doubled compared to the same period in 2017. The main reason for the higher delays during summer of 2018 has been attributed to air traffic control (ATC) experiencing unplanned staffing issues [3].

The issue when there are more flights intending to enter an airspace than there is capacity to safely control these flights, is generally referred to as a demand-capacity imbalance. Air Traffic Flow Management and its European extension Air Traffic Flow and Capacity Management (ATFCM) has as its primary objective to plan and implement measures for demand-capacity balancing [4]. The measures can be categorized into three main classes: (1) Optimization of available capacity; (2) Shifting the demand into other areas; (3) Regulating the demand. Within the last class only the measures known as ATFCM regulations are considered in this study. Such a measure involves assigning a pre-departure delay on ground for each flight affected by the regulation.

Email addresses: a.doutsis@student.tudelft.nl (Author MSc student: A. Doutsis), m.mitici@tudelft.nl (TU Delft supervisor: M. Mitici), rasoul.sanaei@dlr.de (DLR supervisor: R. Sanaei)

Incentivised by the delays experienced during the summer months of 2018, the focus of this paper is the tactical phase of ATFCM operations. Specifically, this study is aimed at bringing forth improvement on process planning during this phase by having a better understanding of the evolution of measures for regulating the demand under an imbalance. This is achieved through a predictive model with which characteristics, that relate traffic complexity with performance of planned measures over an Area Control Center (ACC) or Upper Area Control (UAC), are predicted for different forecast horizons.

The above high-level objective is split into two main research objectives. The first objective is to create a model with which the number of new, changed and cancelled regulations, total activation time and duration of future regulations can be predicted. The second objective is to research and implement a methodology with which the maximum prediction horizon for different ACCs/UACs can be estimated.

In recent years, machine learning algorithms have shown very good performance in making predictions. Extensive research in the available literature indicated that there is a lack of studies that utilize machine learning to predict the characteristics of European ATFCM regulations. The most similar literature to this research in the last years consists of [5] and [6].

In [5] Liu and Hansen have investigated the problem of predicting the initiation of a Ground Delay Program (GDP) for different forecast horizons. GDPs are the US equivalent to European ATFCM regulations [7]. By utilizing a logistic regression model together with demand, capacity, flight schedules and weather data they conclude that increasing the forecast horizon from 1 hour to 4 hours does not lead to significant increase of the prediction errors.

In [6] Estes et al. investigated the performance of a Random Forest in predicting the average arrival delay caused by a GDP. They consider all the historical GDPs to have occurred at a single location and they are weighted based on the similarity of the traffic and weather conditions at the time of occurrence. The authors found that the Random Forest model together with the weighing scheme resulted in a mean absolute error (MAE) of 11.6 minutes.

Other authors investigated the application of tree-based machine learning models on the task of arrival and departure delay prediction without accounting for the cause of the delay. In [8] Thiagarajan et al. investigated a variety of tree-based models for the task of predicting the value of arrival and departure delays of flights. As input the authors used US airline on-time-performance data and weather data. They found that the best results were obtained from the Extra-Trees model and the second best from Random Forests. Similarly, in [9] Manna et al. utilized Gradient-Boosted trees for predicting arrival and departure delays. Using as input features the day of week, airline, origin/destination airport and scheduled departure/arrival times they found a MAE of 7.56 minutes for arrival delay and a MAE of 4.7 minutes for departure delay.

The Network Manager (NM), as the head of the collaborative decision making process in ATFCM has an overview of the traffic situation over the European ATM Network and utilizing systems such as SIMEX and PREDICT [4] can run simulations to predict and asses the impact of the future measures for demand-capacity balancing. However, for these systems to work a holistic overview of the network is needed. That requires full knowledge on sector demand, forecasted demand and declared capacity. As an external stakeholder of the system this information is rarely available, as a result of this the predictions will have to utilize other means than demand and capacity.

The best proxy information that is publicly available consist of the ATFCM Notification Messages (ANMs). The list of ANMs is publicly available through Eurocontrol's Network Operation Portal (NOP). Adding, changing or cancelling planned ATFCM regulations cause the list of ANMs to be updated to reflect the current situation.

Because of the structuring of European airways and the locations of the busiest airports, some ACCs/UACs are more active than others from a regulatory point of view. As an example, consider the ACCs/UACs in FABEC Functional Airspace Block. In these sectors some of the busiest airports are located and at the same time the most flown routes pass over these areas. Due to this diversity in regulatory dynamics over different air sectors there is a need to have a methodology with which the maximum forecast horizon over different ACCs/UACs can be assessed.

A possible idea for estimating the forecast horizon is to set it equal to the mean return interval of regulations. That is, the average time between the start times of two consecutive regulations. However, Bunde et al. [10] have proved that the mean return interval does not account for long-range dependant behaviour.

The most well-known continuous-time stochastic process that shows long range dependence, is the self-similar process of fractional Brownian motion (fBm) and its increment fractional Gaussian noise (fGn) developed by Mandelbrot and van Ness [11]. In this type of process, the level of dependence is quantified through the Hurst exponent, named after the British hydrologist Harold Edwin Hurst.

The Hurst exponent H takes values between 0 and 1. When $H = 0.5$ the increments of the process are uncorrelated, and the overall process is similar to a random walk [12]. When $0 < H < 0.5$ the increments of the process are negatively correlated, with the anti-correlation getting stronger as $H \to 0$. When $0.5 < H < 1$ the process exhibits long range dependence and the increments of the process are positively correlated [12], with the strength of the correlations increasing as $H \to 1$.

Karagiannis et al. [13] offer an overview of all the methods available to estimate the Hurst exponent. However, as it can be seen in [12, 14, 15, 16] the rescaled range method is the one most commonly used. This method is the original procedure proposed by Mandelbrot and Wallis [17].

Wang et al. [16] utilized the rescaled range method to estimate the Hurst exponent of air traffic flow time series constructed at different time-scales. For the series constructed with the smallest time-scale (10 minutes) they found $H = 0.72$. As the time-scale was increased they found a decrease of the Hurst exponent, with $H = 0.64$ for a time-scale of 30 minutes. They conclude that as the time-scale of observation increases the process start becoming more chaotic.

Molino-Minero-Re et al. [14] have proposed a method with which the Hurst exponent can be estimated as a function of time and time-scale, the Time-Scale Local Hurst Exponent (TSLHE). The method is based on a sliding window with varying width (time-scale). At the end of the process a matrix, whose element H_{ij} indicates the Hurst exponent at time $t = i$ and at the j^{th} time-scale, is obtained. After averaging the matrix to remove the time-scale dimension they utilize the average Hurst exponent as a function of time to detect structural changes in seismic time series.

Qian and Rasheed [18] tested the hypothesis that forecasting time series with Hurst exponent higher than 0.5 leads to lower forecasting errors. After forecasting 30 series with $H > 0.6$ and 30 series with $H \sim 0.5$, using a neural network, they found that the error metric was lower for the set of series with high Hurst exponent compared to the other set. As a final step they ran a Student's t-test with the null hypothesis that the mean error metric for both sets is equal. The p-value for the test resulted to be $7.029 \cdot 10^-10$ leading to rejection of the null hypothesis.

The hypothesis is that the prediction horizon can be estimated from the time-scale at which the increments of the process become uncorrelated. In this way one aims to find the furthest point in time in the future, at which the present information has predictive power. As shown in [18], when the Hurst exponent of the series is bigger than 0.5 the time series can be predicted with lower error rates. From the findings of [16], it can be expected as the time-scale increases the Hurst exponent decreases. Finally, using the TSLHE procedure [14] the average Hurst exponent as a function of time-scale can be obtained.

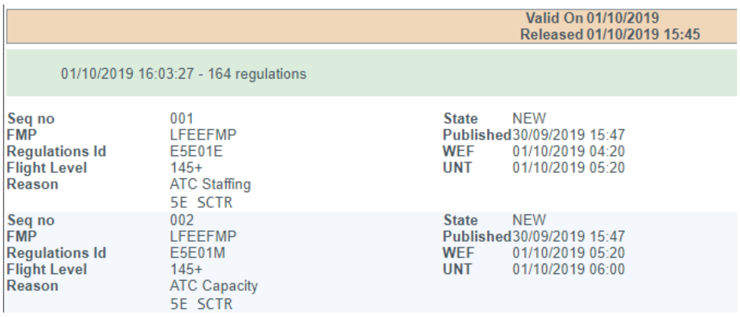
In section 2, the datasets that were used for the purposes of this study are described. Given the two main objectives of this research the methodology and results sections are split into two sections, respectively. In section 3, the methodology related to the problem of determining the prediction horizon is given. This section then is followed by section 4 where the results of the prediction horizon for selected ACCs/UACs are given. In section 5 the methods and steps involved in developing the predictive model are discussed. This is then followed by section 6 where the results of the experiments with the predictive model are given. In section 7 the results related to both research objectives are discussed and interpreted. After the conclusions given in section 8, in section 9 potential improvements together with an application of the predictive model on flight operations are proposed.

2. Dataset description

As it has been discussed in section 1, the data that is used to reach the objectives of the research consists of the lists of ANMs. ATFCM is applied on four different time horizons. This involves a planning process for each calendar day, that for the purposes of the ATFCM process will be referred to as "Day of Operations" or D_{ops}. During the tactical phase (on D_{ops}) the initial plans are evaluated in real-time and adjustments are made so that the implemented measures are at the bare minimum to solve the problem [4].

2.1. Tactical ATFCM data

During D_{ops} the list of ANMs is obtained from the Network Operations Portal (NOP). Throughout the day the regulations are updated as needed to manage the traffic load. A screenshot of the list of ANMs can be found in Figure 1. Figure 1 starts with the release date and time of the list of regulations. Each time a regulation is added changed or cancelled the release time is updated. This is then followed by the date and time a user accesses the list of regulations and the total number of regulations.



Valid On 01/10/2019
Released 01/10/2019 15:45

01/10/2019 16:03:27 - 164 regulations

Seq no	001	**State**	NEW
FMP	LFEEFMP	**Published**	30/09/2019 15:47
Regulations Id	E5E01E	**WEF**	01/10/2019 04:20
Flight Level	145+	**UNT**	01/10/2019 05:20
Reason	ATC Staffing 5E SCTR		
Seq no	002	**State**	NEW
FMP	LFEEFMP	**Published**	30/09/2019 15:47
Regulations Id	E5E01M	**WEF**	01/10/2019 05:20
Flight Level	145+	**UNT**	01/10/2019 06:00
Reason	ATC Capacity 5E SCTR		

Figure 1: Screenshot of ATFCM regulation messages obtained from Network Operations Portal.

The following rows in Figure 1 contain the notification messages for the planned regulations. Each ANM starts of with the **State** indicating the state of the regulation. The **State** field can take any of the following values: *NEW*, *CHANGED* and *CANCELED*

This is then followed by the relevant **FMP**. The first 4 letters of which correspond to the ICAO codes for the ACCs/UACs, giving the geographical location for applicability of the regulation. The **Flight level** field indicates within the location of applicability of the regulation which flight levels (FLs) will be regulated. The **Flight level** field can take any of the following values:

- All FLs- Indicated by *ALL*
- Range of FLs - e.g. *045 - 190*, flight levels from FL045 to FL190
- Upper boundary - e.g. *200-*, all flight levels bellow FL200
- Lower boundary - e.g. *300+*, all flight levels above FL300

There are three different times associated with each regulation. Firstly, there is the time that the regulation was announced to the stakeholders of the system, **Published**. This field will be referred to for convenience as **PUB**. If a regulation is changed or cancelled, this time is also updated. Next there is the time when a regulation is planned to become effective **WEF** and also the time up until it is planned to be in effect **UNT**. For regulations that are cancelled these field are empty and only the **PUB** field is available.

Based on the above regulation times two new fields can be calculated. The duration of the regulation **DUR** is obtained from **UNT**−**WEF**. For cancelled regulations, because the **WEF** and **UNT** fields are not available the duration is set to zero. The activation time **ACT** of the regulation can be obtained from **WEF**−**PUB**. The activation time can take any of the following values:

- Cancelled regulations: Zero values
- New regulations: Only positive values
- Changed regulations: Both negative and positive values. A negative value occurs when a regulation is amended after being in effect

Finally, the last field of interest is the **Reason** for the regulation. This field can take any of the 14 pre-defined reasons from the Network Manager and they are given in Annex 5 of [19].

Typically, the first regulations of the day are published pre-tactically from the day before D_{ops}. This can be seen by comparing the **PUB** fields of the two regulations shown in Figure 1 with the release time of the list of regulations. As the day progresses the regulations are updated so that the tactical phase objectives can be reached. This is illustrated in Figure 2. The two time series shown in Figure 2 represent the total count of non-cancelled regulations in EDUU (Karlsruhe UAC). The blue series is constructed from the list of regulations released on the morning of 5^{th} of April 2019 and the orange one is constructed from the final list of regulations collected for that day. As it can be seen from this plot what was planned initially and what actually happened in the context of regulations deviate considerably.

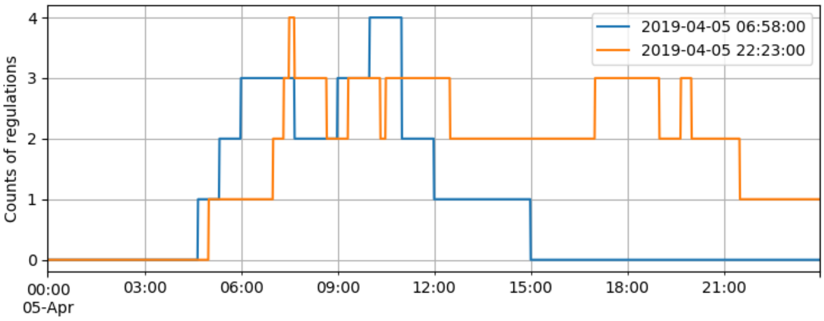


Figure 2: Time series of the counts of non-cancelled regulations in EDUU. The legend indicates the release time of the list of regulations.

In order to capture the evolution of the list of ANMs, a system had been set up by the ATFM group of the DLR Air Transportation Systems. Roughly every ten minutes, this system would access the NOP, extract the ANMs and store the data in a sheet within an Excel file. Each of these sheets is referred to as a snapshot of D_{ops}. After pre-processing, which will be discussed in subsection 5.3, this data will be used in order to train and test the predictive model. In this manner data on the following months were available

- April 2019 - The 6^{th}, 7^{th}, 13^{th} and 14^{th} of the month were missing.
- May 2019 - The 6^{th} and 22^{nd} of the month were missing
- June 2019 - The 19^{th} of the month was missing
- July 2019 - No missing data

As indicated in Figure 2, what is initially planned with what actually happens at the end of the day differ. In order to have the a view over the actual regulatory situation, besides the 10 minutes snapshots, the last snapshot of D_{ops} were collected post-operationally. In the NOP a public user has the ability to access the list of regulations up to 40 days before the current day. This source of data is used for estimation of the prediction horizon of an ACC/UAC. The reason for this is that it reflects what actually happens on D_{ops}, while at the same time is not affected by the limitations of the process with which the 10 minute snapshots are collected. In this way the last snapshots of each of month from March up to and including June 2019 were available, with no missing days.

3. Methodology - Prediction Horizon

The basis for the proposed method is to use the Hurst exponent as an indicator of predictability of a time series. By using the algorithm presented by Mollino-Minero-Re et al. [14], one can obtain the Hurst exponent as a function of time and time-scale. After obtaining this result it is possible to remove the time dimension through averaging to obtain the average Hurst exponent as function of time-scale. The hypothesis then is that the time-scale at which the process becomes uncorrelates is also the forecast horizon. In subsection 3.1 the method to estimate the Hurst exponent is described. In subsection 3.2 the verification steps undertaken to establish the ability of the used method in determining the expected values for the Hurst exponent, are discussed. In subsection 3.3 and subsection 3.4 the implementation of the Time-Scale Local Hurst Exponent is discussed. This section is concluded with subsection 3.6 where the procedure for verifying the value of the maximum prediction horizon is described.

3.1. Rescaled range method

The rescaled range (RS) is a statistic used by the hydrologist Harold Edwin Hurst to study the optimal sizing of water reservoirs. Consider a time series $\{X_t\}$ $t \in 0, 1, 2, ..., N$. A new series Y_t is created as follows

$$Y_t = X_t - \overline{X} \quad \text{for} \;\; t = 0, 1, 2, ..., N \tag{1}$$

where $\overline{X}$ is the sample mean of $\{X_t\}$. $\{Y_t\}$ describes the deviations of the series X_t from its mean $\overline{X}$. Based on $\{Y_t\}$ another series Z_t is created by taking the cumulative sums of Y_t, that is

$$Z_t = \sum_{i=0}^{t} Y_i \;\; \text{for} \;\; t = 0, 1, 2, ..., N \tag{2}$$

as such $Z_0 = Y_0$, $Z_1 = Y_0 + Y_1$ and so on. Finally the range R is defined as

$$R(N) = \max_{0 \leq t \leq N}(Z_t) - \min_{0 \leq t \leq N}(Z_t) \tag{3}$$

In order to standardize the range, Hurst divided it by the standard deviation of $\{X_t\}$ to form the rescaled range as

$$RS(n) = \frac{\max_{0 \leq t \leq N}(Z_t) - \min_{0 \leq t \leq N}(Z_t)}{\sqrt{\frac{1}{N}\sum_{i=0}^{N}(X_i - \overline{X})^2}} \tag{4}$$

According to Mandelbrot and Wallis [17] the RS statistic shows the following assymptotic relationship

$$\lim_{N \to \infty} RS(n) = cn^H \tag{5}$$

where n is the length of the time series, c is a constant and H is the Hurst exponent. Equation 5 can be linearized through a logarithmic transformation to obtain Equation 6.

$$\log\left(RS(n)\right) = \log(c) + H\log(n) \tag{6}$$

Based on Equation 6, the Hurst exponent can be determined by: (1) Calculating the RS for different values of n; (2) Plot the $\log\left(RS(n)\right)$ over $\log\left(n\right)$; (3) Fit a least-squares line over the resulting points and obtain the value of H from the slope of the fitted line. Typically the first step in the procedure is performed by splitting the overall series into disjoint sub-series of equal length. This is illustrated in Figure 3. Starting off with the original series of length n, the RS for this length is calculated. Then the series is split into two equal halves. On each halve the RS is calculated and the RS for a series length $n/2$ is obtained from the average of the two. The process is repeated with three sub-series of equal length and so on up until the lowest sub-series length is reached. The concepts of range and standard deviation are inapplicable for a single number, thus the smallest possible sub-series length is two.

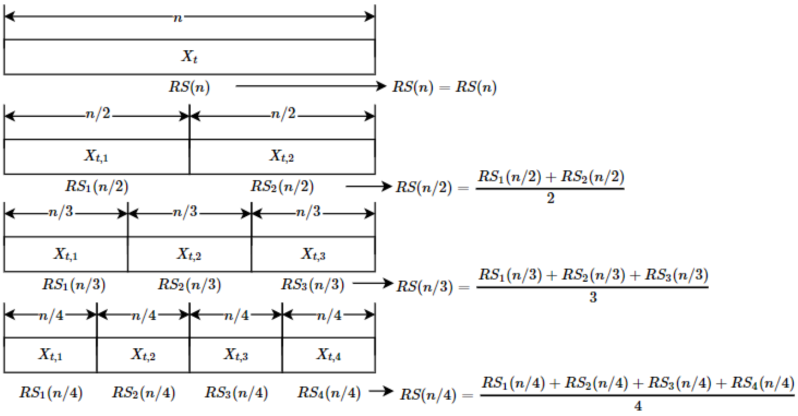


Figure 3: Illustration of the process of calculating the rescaled range of a time series for varying lengths, based on disjoint intervals of equal length.

3.2. Verification of the Hurst exponent estimation

To verify that the procedure for estimating the Hurst exponent was implemented correctly two experiments were conducted. Since the value of H for a series composed from uncorrelated increments is expected to be 0.5 [15], for the first experiment synthetic series were generated by sampling a N(0,1) distribution. The series were constructed for lengths from 2^5 up to 2^{13}. For each series length 100 series were constructed, the Hurst exponent was estimated on each of them and the results are averaged.

For the second experiment, the performance of the implemented method was checked against series with varying values for the Hurst exponent. For generating series with different values for H the Python package "*fbm*" [20] was used to generate series of fractional Gaussian noise (fGn) and fractional Brownian motion (fBm). The algorithm used with in it to generate series of fGn has been proposed by Davis and Harte in [21]. A mathematical description of the procedure implemented can be found in [22, p. 15-17]. The fBm process is obtained by taking the cumulative sum of the fGn series. Similarly to the first experiment, for each value of the Hurst exponent from 0.05 up to 0.95 with a step of 0.05, 100 such series were generated. The value of H was estimated with the rescaled range method and the results are averaged in the end. The results of these verification steps are given in subsection 4.1.

3.3. Time-Scale Local Hurst Exponent (TSLHE)

The procedure for determining the Hurst exponent as function of time and time-scale has been proposed in [14]. This procedure is based on a sliding window, where the window length WL, represents the time-scale of observation. By sliding the window one sample at a time the Hurst exponent is estimated within the sliding window using the rescaled range method. Consider a time series $\{X_t\}\ \ t \in 1, 2, ..., n$.

1. The time-scales $WL_i\ \ i \in 1, 2, ..., k$ at which the series will be analyzed are selected. From a procedural point of view the maximum time-scale WL_k can be bigger than the length of the series $\{X_t\}$, however practically this offers no advantage as such the $WL_k \leq n$. For the minimum time-scale WL_1, the authors of [14] suggest $WL_1 \geq 8$. The reason for this is that at this time-scale there are 3 points for estimation of H.

2. In order to obtain a value of H for each element of the series $\{X_t\}$, the series has to be padded. For the padding strategy the authors of [14] used symmetric reflection of the $\lfloor WS_i/2 \rfloor$ initial and end samples. For the purposes of this research, because of the daily seasonality in the data, it was deemed more appropriate to apply a circularization strategy. In this way, to create the new padded series $X_{pad,m}\ \ \ m = 1, 2, ..., n + WL_i - 1$, at the beginning of $\{X_t\}$ the last $\lfloor WL_i/2 \rfloor - 1$ observations of $\{X_t\}$ are inserted and at the end of $\{X_t\}$ the first $\lfloor WL_i/2 \rfloor$ observations of $\{X_t\}$ are appended. The procedure is illustrated in Figure 4

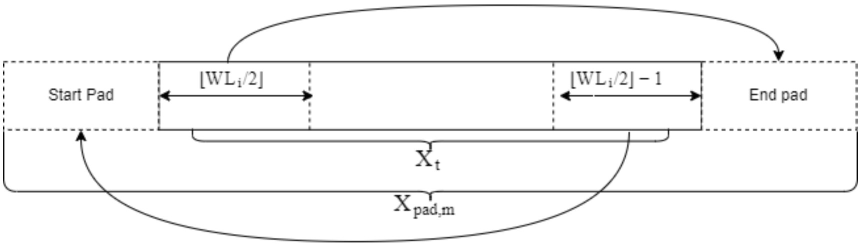


Figure 4: Circular padding strategy, used in the calculation of the Time-Scale Local Hurst Exponent

3. The sliding window of width WL_i is placed at the begining of the padded series. Within this window H is estimated using the rescaled range method. Then the window is slid one sample forward and the estimation repeated up until the window reaches the end of the padded series. This is graphically depicted in Figure 5.

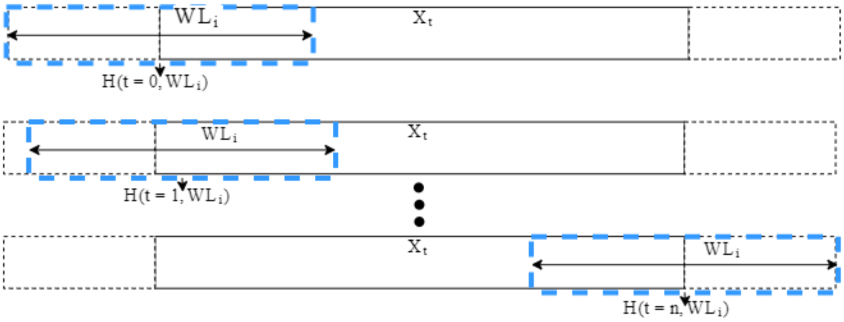


Figure 5: Process of obtaining the Time-Scale Local Hurst Exponent, based on a centered sliding window.

4. At the end of the process a matrix A with dimensions $[n, k]$ is created. Element $a_{i,j}$ of matrix A, represents the level of correlation/anti-correlation or lack of correlation between the increment i of original series with the preceding $j/2$ and succeeding $j/2$ increments of the original series.

After obtaining the resulting matrix averaging is performed over the rows of this matrix to obtain the average Hurst exponent as a function of time-scale. The time-scale at which the process shows randomness ($H \sim 0.5$), which will be referred to as TS_H, then is used as the forecast horizon.

The six following ACCs/UAC were selected for estimation of the prediction horizon: EDGG (Langen ACC), EDUU (Karlruhe UAC), LECM (Madrid ACC), LFEE (Reims ACC), LFFF (Paris ACC) and LFMM (Marseille ACC). These ACCs/UACs were consistently in the top 10 (by number of appearances in the lists of ANMs) of each month from March to June 2019. For this selection, two sets of time series were created. In the first set for each ACC/UAC separate time series where created for each of the 4 months (24 series). This set was used to track how the prediction horizon evolves over the historical data. The results of this step are given in Figure 9. In the second set one series was created for each ACC/UAC by combining all 4 months (6 series). This set was used to determine the overall value of the prediction horizon for each ACC/UAC. These results are given in Figure 8.

All time series were created with a step-size of 15 minutes and by utilizing only the start and end times of non-cancelled regulations. An essential step before applying the TSLHE method on the series, consisted of differencing the series once as shown in Equation 7. The reason for this operation consists on the fact that the created series where found by the Augmented Dickey Fuller (ADF) test and the Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test to be non-stationary due to the presence of a stochastic trend. As it has been discussed in [13] and as it will be seen in subsection 4.1, the rescaled range method results in Hurst Exponents close to and sometimes bigger than one when the input series is non-stationary.

3.4. Singularity of the TSLHE

In the literature that discuss the application of the Hurst exponent, the time series considered are such that at time t and $t+1$ it is highly unlikely to have the same value. Example of this include temperature time series and stock-market prices, which are subject to constant fluctuations. The time series used in this research are such that constant values between t and $t+1$ are highly likely. An example of this was shown in Figure 2. As it can be seen in Figure 2 for the most part the series contains flat plateaus of constant values.

When applying the TSLHE method for such a series, it can occur that for a certain time-scale and location in the series the rescaled range statistic will have to be applied to a part of the overall series where the values are constant. This is problematic since, for a series of constant values both the range R and standard deviation are zero. This results in the rescaled range for this part of the series to be undefined. To mitigate this issue it was decided that for the purposes of the research it would be most appropriate for such intervals to correspond to Hurst exponent of 1.

The reasoning behind this lies in the physical interpretation of the Hurst exponent, where $0.5 < H < 1$ represents positive correlation between the increments of the series. As $H \to 1$ these correlations become stronger. A series of constant values can be considered as a singular number and a singular number is always fully correlated with it self, having a Pearson correlation coefficient of 1.

3.5. Comparison of the prediction horizon with the mean return-interval

In order to verify that the procedure proposed in subsection 3.3 behaves as expected the maximum prediction horizon determined from the TSLHE procedure is compared against the mean return interval. It has been proven in [10], that the mean return interval is invariant to the long-range dependency of the process. As such using the mean return interval for the maximum prediction horizon will not account for the true behaviour of the process. Nevertheless, the mean return interval can be used as an indicator of the dynamics of the regulations in the area of study. As the rate of regulations (inverse of mean return-interval) increases the ACC/UAC can be considered more dynamic and unpredictable. As a result of this, a shorter return interval is expected to correspond to a shorter forecast horizon.

In order to test for this behaviour, after selecting the ACCs/UACs of interest, separate time series where created for each ACC/UAC for the months of March, April, May and June 2019 using the last list of ANMs. For each ACC/UAC and month the maximum prediction horizon TS_H and the mean return interval were calculated. To obtain the regulation return intervals the following procedure was done:

1. Starting with the time series of the total count of non-cancelled regulations over an ACC/UAC, $\{C_t\}$ for $t = 0, 1, ..., n-1$, the first difference was applied to $\{C_t\}$ to create a new series $\{\nabla^1 C_{t'}\}$, where

$$\nabla^1 C_{t'} = C_{t'} - C_{t'-1} \quad for \quad t^{'} = 1, 2, ..., n-1 \tag{7}$$

2. All positive values in $\{\nabla^1 C_{t'}\}$ indicate the start of r regulations, with $r > 0$. The times at which this values occur are called the arrival times of the regulations.
3. Finally the return intervals are obtained by the time difference between two consecutive arrival times.

3.6. Verification of the prediction horizon

To verify the value obtained for the maximum prediction horizon obtained from the TSLHE method, TS_H is compared to a similar methodology based on the largest Lyapunov exponent (LLE) of a chaotic system. Consider a deterministic system composed of g states, with $g > 0$. Such a system is chaotic if, small changes in the initial conditions lead to an unpredictable evolution of the states. If we consider two initial conditions of infinitesimally small distance $\delta_0 \in \mathbb{R}^g$, the Lyapunov exponent quantifies the rate of divergence $\delta(t) \in \mathbb{R}^g$ between the two initial conditions [23]. As it has been shown in [23] the largest Lyapunov exponent can be estimated from the following relationship, where λ is the LLE

$$\|\delta(t)\| \cong \|\delta_0\| e^{\lambda t} \tag{8}$$

When the system has a positive LLE then the concept of a prediction horizon T_L exists and after it all predictions degrade [23]. To calculate T_L the following relation is given [23]

$$T_L = \frac{1}{\lambda}\left(\ln \frac{a}{|\delta_0|}\right) \tag{9}$$

where a represents the tolerance for the predictions. As it has been shown in [24] when considering time series, the tolerance and initial distance can be omitted to obtain the prediction horizon for this series from the inverse of the LLE, $T_L = 1/\lambda$. An algorithm to calculate the LLE has originally been proposed by Rosenstein et al. [25]. The basis of this algorithm is reconstructing the phase space of the dynamical system. The phase space of the example system is the g-dimensional space that contains all the state vectors indexed by time.

If the time evolution of the g states of the system are available, the phase space can be reconstructed perfectly. However, using Takens Embedding Theorem [26] this can also be done by using only a single time series $\{X_t\}$ for $t = 1, 2, ..., n$ of the system. This is achieved by selecting an appropriate embedding dimension d and lag τ then the reconstructed phase space vectors $X_{R,t}$ can be obtain as follows [27]

$$X_{R,t} = \left[X_t, X_{t+\tau}, X_{t+2\tau}, ..., X_{t+(d-1)\tau}\right] \quad \text{where} \quad t = 1, 2, ..., n-(m-1)\tau \tag{10}$$

To select the appropriate lag τ, the most common method has been proposed by Fraser and Swiney in [28]. In this method the mutual information (MI) between the original series $X(t)$ and a lagged version of it by τ is computed using the following equation

$$MI(\tau) = \sum_{i=1}^{n} p(X_i, X_{i+\tau}) \log_2 \left[\frac{p(X_i, X_{i+\tau})}{p(X_i)p(X_{i+\tau})}\right] \tag{11}$$

where $p(X_i, X_{i+\tau})$ is the joint probability mass function of $\{X_t\}$ and $\{X_{t+\tau}\}$, $p(X_i)$ and $p(X_{i+\tau})$ are the marignal probability mass functions and the $\log_2$ is used to obtain the MI in units of bits. The value of τ to be used for the reconstruction then is the first local minimum of the curve $MI(\tau)$ over τ.

To determine the embedding dimension the method of false nearest neighbors is used [29]. A phase space that is reconstructed in the optimal dimension d^* is a one to one mapping of the original phase space. As such neighbours of the original phase space remain neighbours in the reconstruction. When the embedding dimension is smaller than d^*, false neighbours will appear in the reconstructed phase space due to projecting a higher dimensional object to a lower dimension. Within a dimension d the Euclidean distance is computed for each point $X_{R,i}$ and its nearest neighbor $X^n_{R,i}$ as follows.

$$R_i(d) = \left\|X_{R,i} - X^n_{R,i}\right\|^2 \tag{12}$$

The points are considered false neighbors if the following relation holds true

$$\sqrt{\frac{R_i(d+1) - R_i(d)}{R_i(d)}} > D \tag{13}$$

where D is a distance threshold value. The embedding dimension to be used for the reconstruction then is the first dimension that has a fraction of false nearest neighbors bellow 10%.

Finally, after obtaining the values for d and τ the phase space $X_{R,t}$ is reconstructed. For each point i, in the reconstructed space, the nearest neighbour to this point i^* is determined. The condition for two points to be nearest neighbours is that they should be separated by the mean period of $\{X_t\}$. Equation 8 can be rewritten as follows

$$\ln\left(\frac{|\delta_{i,t}|}{|\delta_{i,0})|}\right) = \lambda t \tag{14}$$

where $\delta_{i,t}$ represents the distance between point i and its nearest neighbor i^* at time t and $\delta_{i,0}$ is the initial distance between the two. By calculating the right hand side of Equation 14 for different values of t and plotting the results, the LLE can be determines as the slope of the line that best fits the points. In order to have a reliable estimate for the LLE the right hand side of Equation 14 is calculated for all the pairs (i, i^*) and the results at a time step t are averaged over all pairs.

4. Results - Prediction Horizon

In this section the results of applying the proposed method for determining the maximum prediction horizon are given. In subsection 4.1 the results of the verification experiments described in subsection 3.2 are given. In subsection 4.2 the results of the maximum prediction horizon for the six selected ACCs/UACs, together with the changes of this value over the months in the dataset, are given. Finally the results of the verification procedure described in subsection 3.6 are given in subsection 4.3

4.1. Rescaled range method verification

In Figure 6, the results of estimating the Hurst exponent of time series of varying length generated from sampling a N(0,1) distribution are shown. For each series length, 100 series where generated and their estimated Hurst exponents are averaged out to obtain the mean estimated Hurst exponent. As it can be seen from this figure, for small series length the estimated value of H deviates from the expected value of 0.5. As the series length increases the value of the Hurst exponent seems to approach the expected value. Based on these results, to determine the time-scale at which the process start to be random, TS_H, a value of $H = 0.6$ was used as the indicator for the random behaviour.

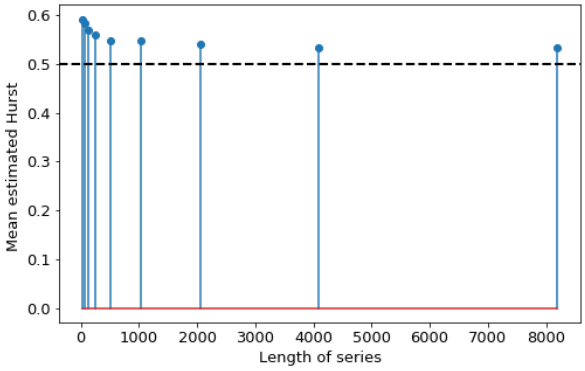


Figure 6: Results of Monte Carlo simulation for estimation of the Hurst exponent on series generated from sampling a N(0,1) distribution.

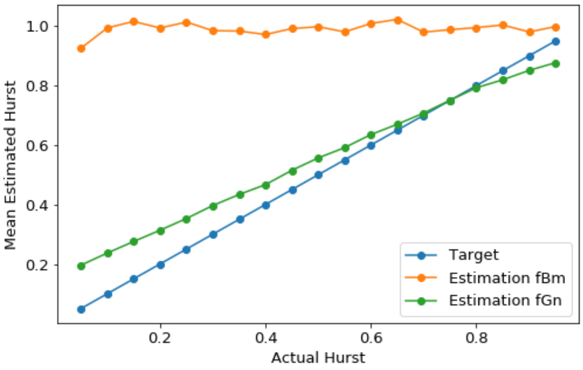


Figure 7: Results of Monte Carlo simulation for estimation of the Hurst exponent on fGn and fBm series. All series were created with a length on 2048 samples.

In Figure 7, the results of estimating the Hurst exponent for fGn and fBm series with varying Hurst exponent are given. For each value of H from 0.05 to 0.95, 100 fGn and 100 fBm series were created. After estimating the value of the Hurst exponent on them, through the rescaled range method, the estimated values are averaged. In Figure 7, the "Target" line represents the ideal line for the estimations.

As it can be seen from this figure for the fBm series the rescaled range method is always estimating values of H close to or even bigger than 1. The reason for this behaviour is that fractional Brownian motion, much like Brownian motion, is a non-stationary process in which the variance of the process is a function of time. Due to this behaviour of the rescaled range method, a first differencing operation was applied on all input series, as discussed in subsection 3.3, to make them stationary.

With respect to the fractional Gaussian noise series, it can be seen from Figure 7 that the estimated values of H deviate from the ideal line. For values of $H < 0.75$ the rescaled range method overestimated the value of H, meanwhile for $H > 0.75$ the values of H are underestimated.

4.2. Prediction horizon results

In Figure 8, the estimated prediction horizons TS_H for each ACC/UAC selected, are given. After computing the TSLHE matrix for the total count of non-cancelled regulations time series over the six ACCs/UACs for the course of 4 months, the matrices are averaged to obtain the six curves shown in Figure 8. Using a threshold value of $H = 0.6$ to indicate randomness the time-scale when this occurs TS_H represents the maximum prediction horizon. As the time-scale increases the average Hurst exponent of the input series decreases as it was also observed in [16]. For half of the cases shown in Figure 8, the curve stabilises at a value of $H = 0.5$ for time-scales bigger than 16 hours. Meanwhile for the remaining ones for time-scales larger than 16 hours the curve fluctuates between $0.4 \leq H \leq 0.5$. The smallest TS_H is observed for EDUU, which is one of the most regulated sectors in Europe, with a value of 4.77 hours. The highest value is observed on LFMM with $TS_H = 7.49$ hours. For all ACCs/UACs, with the exception of LFFF, the mean return interval is smaller than TS_H. The difference between mean return interval and TS_H varies 0.56 hours for LFFF up to 2.42 hours for EDUU.

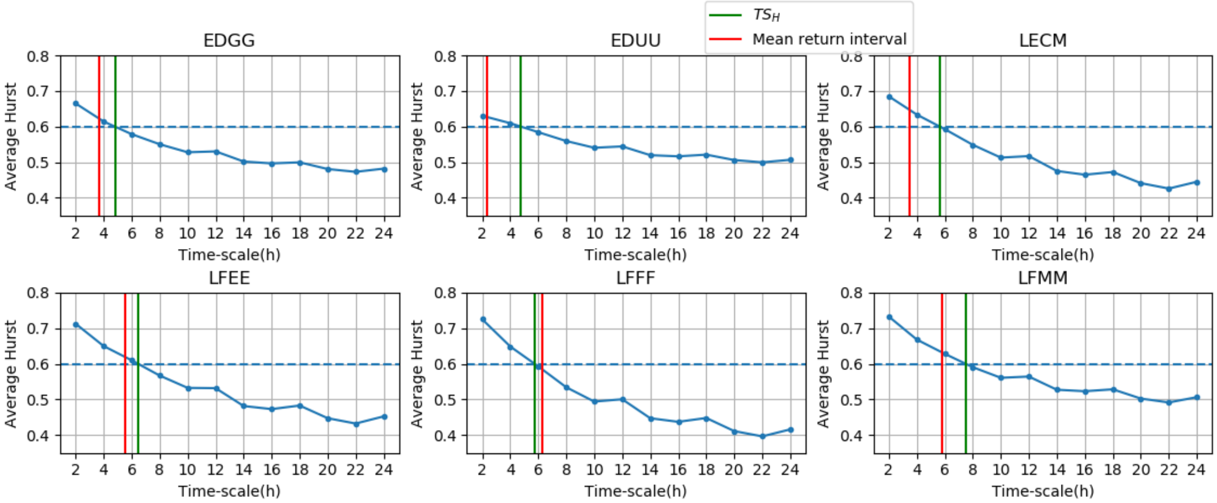


Figure 8: Maximum prediction horizon TS_H results and the mean return interval of regulations over the course of 4 months for each of the selected ACCs/UACs

In Figure 9 the time-scale indicating the start of randomness TS_H and the mean return interval of the regulations for each month are given. When the value for TS_H is negative in Figure 9, it represents the case when averaging the TSLHE matrix to obtain the curve of average H over the time-scale all points of the curve where bellow the threshold for randomness ($H = 0.6$). For the most part it can be seen the curves of TS_H and mean return interval are changing in a similar manner when going from March to June.

To obtain a better understanding of these results consider Figure 10 and Figure 11. In Figure 10 the total number of regulations for each ACC considered is given and grouped by month. Looking at this figure it can be seen that the top three ACC by number of regulations are EDUU, LECM and EDGG. However, when looking at the value for the TS_H, in Figure 8, for LECM it can be noticed that it is considerably higher when compared to the TS_H obtained for EDUU and EDGG. To explain this behaviour consider Figure 11. In Figure 11 the total amount of time without any active regulations is given. From this figure it can be seen that in the case of LECM there is a lot of time spent without regulations when we compare it to EDUU and EDGG. From this discussion it can be inferred that regulations happen more infrequently in LECM, but when they occur they are characterized by burst of regulations occurring simultaneously. This suggest an inverse relationship between the frequency of occurrence of regulations and the maximum prediction horizon.

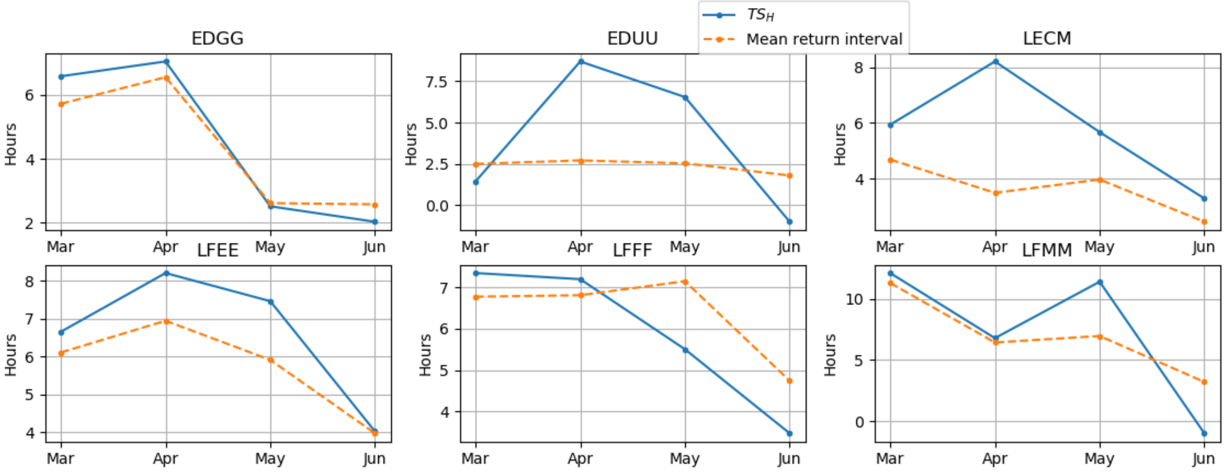


Figure 9: Maximum prediction horizon TS_H and mean return interval over March, April, May and June 2019. Negative value for TS_H indicates that the curve of average Hurst exponent over time-scale was always bellow the threshold of 0.6

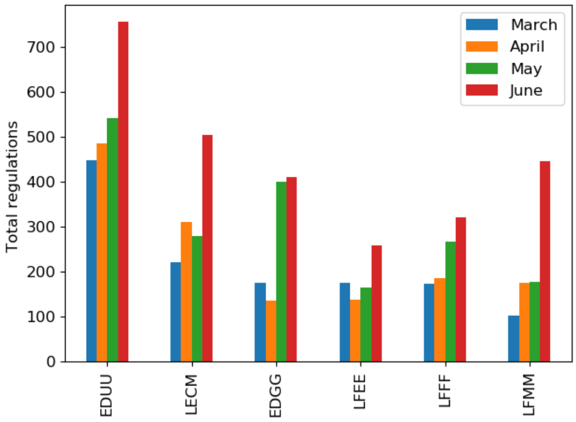


Figure 10: Total number of regulations in the selected ACCs/UACs

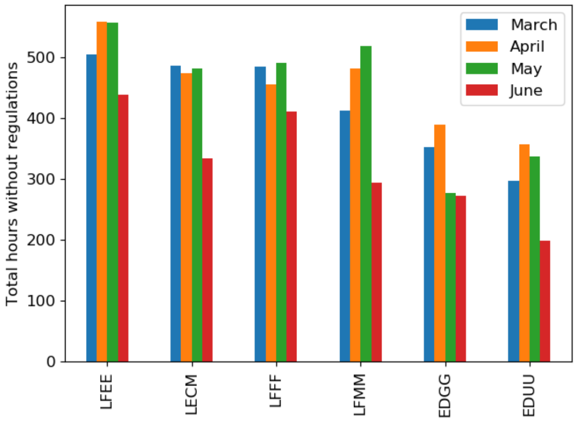


Figure 11: Total amount of time in hours spent without regulations for the selected ACCs/UACs.

4.3. Prediction horizon verification

For verification of the values obtained for the maximum prediction horizon TS_H, the case of EDUU was selected to calculate the prediction horizon T_L using the inverse of its largest Lyapunov Exponent. Using the same time series which was used to generate the results in Figure 8, the phase space of the dynamical system was reconstructed using the count time series of non-cancelled regulations in EDUU over March, April, May and June. Using the Matlab function "*phaseSpaceReconstruction()*" the embedding dimension d was found to be $d = 2$ and the lag τ was found $\tau = 7$.

With these parameters fixed using the function "*lyapunovExponent()*" in Matlab, the logarithmic divergence of all points i in the reconstructed phase space and their neighbors i^* (right hand side of Equation 14) was computed for values of $t = 1, 2, ..., 50$. For each time step this logarithmic divergence is averaged over all pairs (i, i^*) to give the average logarithmic divergence. This function takes as an additional input the sampling frequency. The inverse of the sampling frequency (step-size) is used to scale the values of the average log divergence so that when inverting the LLE the result is in units of time. By setting the sampling frequency to 1 the inverse of the LLE returns T_L in time steps.

The results of this process are shown in Figure 12. The plot shown in Figure 12 starts of with a linear increasing region of the log divergence, followed by a transitioning region after which it flattens out. In [25] and [23] the flatting behaviour is attributed to the fact that the divergence of two neighbors can not exceed the size of the phase space. Following the recommendations of [25], the region between time steps 1 up to and including 14, where the divergence seems to be linearly increasing, was selected for performing the line fitting. The slope of this line estimates the LLE of the system. Using the inverse of the value shown in Figure 12 the prediction horizon $T_L = 18.49$ time steps. Because the time series is created with a step-size of 15 minutes, $T_L = 4.62$ hours.

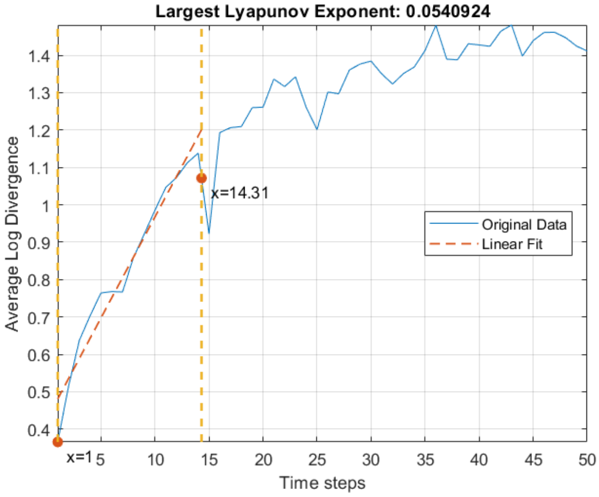


Figure 12: Estimation of the largest Lyapunov exponent for the total count of non-cancelled regulations in EDUU

5. Methodology - Predictive Model

In this section the steps taken in the development of the predictive model are described. In subsection 5.1 the workings of the Random Forest algorithm are described. In subsection 5.2 the problem that has to be solved is defined together and the target predictions are given. Then in subsection 5.3 and subsection 5.4 the steps with which the data is processed to create the input variables to the model are discussed. In subsection 5.5 and subsection 5.6 the experiments with which the best input variables and hyper-parameters were selected, are discussed. This section is concluded with subsection 5.7 where the procedure to validate the result of the maximum prediction is given.

5.1. Random Forests

Based on the literature review it was observed that the majority of papers investigated, utilized Random Forests or found that ensembles of homogeneous learners outperformed other models in regression problems. For this reason it was decided to investigate the performance of a Random Forest on the problem at hand.

The elementary working unit of a Random Forest is the Decision Tree. Consider the training input vectors $\overrightarrow{x_i} \in \mathbb{R}^f$ where $i = 1, 2, ...s$, with s the number of training samples and f the features or variables, and the expected output vector $\overrightarrow{y_i} \in \mathbb{R}^p$ where p is the number of target predictions. A decision tree, starting from its root node, aims to iteratively partition the number of samples s into subsets based on a binary decisions. This decisions are made by selecting an appropriate feature f_j and a threshold value for it t_{f_j} and splitting the training data into left and right sets, S_{left} and S_{right}. In regression problems, the criteria for choosing the parameters for the decision consist of selecting (f_j, t_{f_j}) so that the mean square errors between mean expected output and the expected outputs is minimized on both S_{left} and S_{right} [30].

In a Random Forest model, several decision trees are built in parallel through sampling with replacement (bootstrapping) over the samples s and also over the features f. The end prediction for regression problems then is determined by averaging the predictions of each individual tree. Averaging contributes to error reduction through reducing the variance between the elementary decision trees [31]. A further decrease in variance comes from sampling over the training instances and the feature space reducing the correlation between individual trees [31]. In order to construct the Random Forest model Pythons Scikit-learn [32] package was used. Given the prevalent usage of this package in machine learning literature it is assumed that the implementation of Random Forest Regression has been verified.

5.2. Problem formulation

During the tactical phase of operations, the list of planned regulations are subject to changes throughout the day in order to meet the objectives of this phase. To obtain an understanding of the behaviour of the ATFCM regulations in an ACC/UAC during the tactical phase it is wanted to have a model through which the evolution of this list of ANMs can be predicted. The target predictions that were selected are listed bellow together with the reasons for selection.

1. Number of *NEW*, *CHANGE* and *CANCEL* regulations - The count of regulations reflects both the dynamic air traffic situation over the ACC and also the performance of the planned regulations at time t. At the same time from t to $t + \Delta t$ planned regulations can be subject to changes and cancellations. Being able to predict the number of new, change and cancelled regulations can offer insights into the reactivity of the decision being made.

2. Activation time of regulations - The activation time can be considered as an indicator of the way decisions are being made during the tactical phase of operations. Very high activation times reflect a very predictable traffic situation in which decisions can be made well ahead of the foreseen imbalance. Low or even negative activation times on the other hand indicate that the traffic patterns during the day of operations are complex as such the decisions have to be made on a short notice.

3. Duration of regulations - Longer lasting regulations are expected to affect a bigger number of flights planning to enter the regulated sector. Depending on the time of the day predicting the duration of regulations can be useful in determining the impact of the regulations on the induced ATFM delay on flights.

With these target predictions fixed, the problem statement was defined as follows: *Given the list of regulations planned for an ACC/UAC at time t, predict the number of regulations, their activation time and their duration at* $t + \Delta t$*.*

5.3. Data processing

The data that was used for training and testing the Random Forest model consisted of the 10 minute snapshots of the list of ANMs obtained from NOP. After pre-processing the raw data which involves transitioning the format shown in Figure 1 into a tabular format where each row corresponds to a regulation and the columns correspond to the fields discussed in subsection 2.1, the snapshots were filtered to contain only the regulations over the ACC/UAC under consideration. Unless otherwise stated for all experiments Karlsruhe UAC (EDUU) was used as it is one of the most active air sectors from a regulatory perspective.

As it was discussed in subsection 2.1, the list of ANMs mainly consists of categorical data with the exception of regulation duration and activation time, respectively **DUR** and **ACT**. Another challenge of the data is that at different time instances the number of regulations can vary, meanwhile the model has to be trained on an input vector of a fixed size. As a result of this the fields of an ANM have to be transformed so that the list of ANMs at time t can be aggregated to an input vector of fixed size.

Given the target predictions stated in subsection 5.2, it was decided that the list of ANMs had to be aggregated through summation. In this way the grand totals of the target predictions at time $t + \Delta t$

will be predicted from grand totals of the list at time t. The notation in the following paragraphs is as follows: Variables in bold represent the variable pre-processing and pre-aggregation; Italicized variables with a subscript i represent the variable post-processing and pre-aggregation; Finally, when the subscript i is dropped the variable is the input feature created after processing and summation.

Time related variables. In order to deal with the time related fields of the ANMs such as **WEF**, **UNT**, **PUB**, the day was discretized into four 6 hour long intervals (00:00 to 06:00, 06:00 to 12:00, 12:00 to 18:00 & 18:00 to 24:00). In this way the continuous time field **WEF** is converted into 4 binary variables $WEF0\_6_i$, $WEF6\_12_i$, $WEF12\_18_i$ and $WEF18\_24_i$. As an example a regulation planned to start at 07:00 will have $WEF6\_12_i = 1$ and the others will be 0. The same is done for **UNT** and **PUB**, resulting in 12 new features.

State features. The **State** field was encoded to form three new binary variables $STATE\_N_i$, $STATE\_CH_i$, $STATE\_CNL_i$. As an example if regulation i is a changed regulation $STATE\_CH_i = 1$ and the rest equal 0

Flight level features. The field **Flight level** can take many different values, if it were be to be one hot encoded this would lead to over 100 new features. In order to avoid the massive increase in dimensionality it was decided to split the airspace into 4 flight level regions. All the last snapshots for each D_{ops} were inspected to find the highest occurring flight level so that the upper boundary can be set there. From this analysis it was found that the highest flight level in the regulation data for the whole network was FL395, with the only exception being LSAG (Geneva ACC) in case of military exercises. For this reason the upper boundary for the flight level regions was set to FL395. In this way **Flight level** was transformed into 4 binary variables $FL395\_295_i$, $FL295\_195_i$, $FL195\_095_i$, $FL095\_000_i$. As an example if regulation i has FL_i = *200-* then, $FL295\_195_i$, $FL195\_095_i$ and $FL095\_000_i$ will equal to 1 and the remaining equal to 0.

Type variables. The ANM field **Reason** can take any value between the 14 regulation reasons given in Annex 5 of [19]. Following a study that had already been conducted by DLR Air Transportation Systems, it was found that the most appropriate grouping of regulations reasons was to have 6 regulation reason categories. Thus the field **Reason** was transformed to the following 6 binary variables ATC_CAP_i, $AERODROME\_CAP_i$, $ATC\_INDUSTRIAL\_ACTION_i$, $ATC\_ROUTEINGS_i$, $WEATHER_i$ and $REST_i$. The first five regulation reasons remain ungrouped and all the remaining types are grouped under REST. As an example if regulation is due to bad weather, $WEATHER_i$ will equal 1 and the remaining will equal to 0.

Tactical or pre-tactical regulations. The first regulations in the list of regulations for each day of operations usually are published one day before. Two additional binary variables were created to convey this information. Based on the **PUB** field and the release date of the list of regulations $PUB_{Dops,i}$ and $PUB_{Dops-1,i}$ were created to indicate whether regulation i was published on the day of operations or on the day before.

5.4. Feature engineering

Up until this point only the features that are directly available from the regulation list at time t have been discussed. As mentioned to create the input vector to the model it was decided to perform summation over the processed features of the list of ANMs at time t. There were two major concerns in performing such a step.

Firstly, in case between two consecutive lists of ANMs at time t and $t + c$ nothing has occurred from a regulation standpoint, summation over the features discussed so far will lead two identical input vectors. The problem that arises then is when splitting the overall dataset to create training and testing sets, where the testing set may overlap with the training set leading to a contamination of the test set with the training samples. In order to prevent this the following features were created to guarantee uniqueness of the input vectors even in the case that the regulatory situation does not change in between consecutive snapshots.

Status of the regulations. For each regulation snapshot at time t it can occur that some of the regulations have already finished, some may be ongoing and some have not started yet. For this reason three new binary variables were created: FIN_i, ONG_i, TO_START_i. These variables describe if regulation i has finished, is ongoing, or has not started yet respectively.

Status related times. The above three status features however are not enough to guarantee uniqueness between consecutive inputs. For this reason the fields **WEF** and **UNT** were combined with the the release time of the list of ANMs to form four new time variables. For regulations that have finished by the release time t, T_PAST_i was created to indicate the time passed in minutes since regulation i finished. For ongoing regulations at time t, T_ELAP_i and T_REM_i were created to describe for regulation i the time elapsed since its start and the time remaining until it finishes. Finally, for regulations that have not started yet, T_TS_i was created to indicate the time until regulation i starts.

Release time variables. Post aggregation of the list of ANMs, 4 more features related to the release time of the list were added as input variables. These are $Curr\_m$, the current month (1-12); $Curr\_d$, the current day (1-31); $Curr\_wd$, the current weekday (0:Mon - 6:Sun); $Curr\_h$, the current hour (decimal hour).

The variables represented so far are considered as the baseline features, they contain all the information available from the list of regulations and also guarantee uniqueness of the input vector at all times. An overview of these input features and their description is given in the first column of Table A.4.

The second point of concern was related to the fact that when the list of regulations is summed to form the input vector, a loss of information could be induced. To mitigate this problem it was considered to add additional features that either describe the statistics of the list of regulations pre-aggregation or include lagged variables that describe the changes in the regulations between t and $t - c$.

Statistical features. For all the non-binary features described so far it was considered to include statistics, such as minimum, maximum, mean and standard deviation before the input vector is created. These non-binary variables include the status related times, duration and activation times of regulations. A summary of these features can be found in the second column of Table A.4.

Lagged variables. The lagged variables are constructed based on differences i.e what changed now compared to k hours ago. This was done post aggregation by comparing input vector at time t with the input at time $t - k$. A summary of these variables can be found in the third column of Table A.4. The naming convention chosen for these variables follows the following format: $\mathrm{L}_k\,\Delta\,\mathrm{input\ var}$. This can be read as lag k hours change in the input variable. For the lags it was decided to only utilize 1 hour and 2 hours.

5.5. Feature selection

An overview of all the features available from the data and the ones constructed is given in Table A.4. In order to determine the best features to be used for the following four experiments were conducted:

1. Using the baseline features from Table A.4 to obtain a baseline estimate of the prediction errors for the five target variables.
2. Using the baseline features and the statistical variables given in Table A.4.
3. Using the baseline features and the lagged variables from Table A.4.
4. Using all of the features listed in Table A.4

To conduct the experiment the 10 minute snapshots for the months of April and May 2019 were used. For the prediction horizon Δt one hour was used. Due to the nature of the data collection process it is unlikely that for a list of ANMs at time t, there is a corresponding list collected exactly at $t + \Delta t$. For this reason when mapping the input to their expected output a tolerance on the prediction horizon has to be considered. In order to have a full utilization of the list of regulations collected over April and May 2019, it was decided to set the prediction horizon tolerance Δt_{tol} to three hours. In this way the predictions are done for one hour into the future and no more than the forecast horizon for EDUU found in subsection 4.2.

Thus, for EDUU 2351 input-output pairs were created. For each experiment above, the dataset was shuffled and 85% of the samples were used for training and 15% for testing. The seed for the random number generator used for shuffling was fixed so that in each experiment the same training and test sets are used. The model was constructed using Scikit-learn package [32] and the hyper-parameters of the model were left to the default values. The performance of each model was evaluated on the test set using root mean square error (RMSE), mean absolute error (MAE) and coefficient of determination (R^2).

5.6. Tuning the model

After feature selection, the model was tuned to further reduce the prediction errors. The tuning was performed in two stages, using the same dataset used in subsection 5.5. Firstly, a screening was performed to determine the most important hyper-parameters and their relevant values so that a parameter grid can be constructed. Secondly, using the constructed parameter grid a grid-search was performed to find the optimal hyper-parameter combination. For each combination the error metric to be minimized is evaluated by applying KFold cross validation, with 10 folds, on the training set. At the end of the process the best parameter combination is chosen and the model is evaluated on the testing set.

5.7. Validation of the maximum prediction horizon

After having a finalized predictive model, different input datasets where generated by increasing the value of Δt from 1 up to 9 hours. The value for the prediction horizon tolerance Δt_{tol} was fixed to be 0.5 hours. For each dataset, using the selected features from the feature selection and the optimal hyper-parameters determined from the grid-search, K-fold validation was used to determine the generalization error of the models for each Δt.

In this technique the dataset is split into K sets, the model is trained on $K-1$ sets and it is tested on the remaining set. In the end of the process for each value of Δt, K models are trained and the test errors are averaged to give a more pragmatic value of the predictive error of the model. For each Δt the K-fold error in terms of RMSE, MAE and R^2 is recorded and the results were plotted.

A representative value for the prediction horizon, stemming from the operational data, was obtained from the value of Δt above which the K-fold errors would start to increase. This prediction horizon is then compared with the maximum prediction horizon TS_H to establish the validity of the TSLHE procedure in determining the prediction horizon.

Finally, the consequence of increasing the value of Δt was that the number of input-output pairs would decrease. In order to exclude the lack of enough training samples as potential cause of an increase of K-fold errors, the full dataset of snapshots recorded over the months of April, May, June and July 2019 were used.

6. Results - Predictive Model

In the following subsections the results related to the development of the predictive model are given. For all the following results the 10 minute snapshots collected during April and May 2019 were used. The prediction horizon Δt is set to 1 hour meanwhile the horizon tolerance Δt_{tol} is 3 hours. Through the mapping procedure for EDUU, 2351 input output samples are used for all experiments. In order to understand the values of the performance metrics, statistics on the value to be predicted are given in Figure 13. From Figure 13, it can be noticed that the number of new regulations is always bigger than 0, indicating that EDUU had one or more regulations in all the available data. As a consequence of this, the total activation time and duration are also positive. On the contrary the number of changed and cancelled regulations may be zero.

6.1. Feature selection results

An overview of all the features available from the data and the ones constructed is given in Table A.4. The results of the experiments discussed in subsection 5.5 are shown in Table 1. For each experiment the error metrics of the test set for all target variable are given. In addition to that on the right of each experiment the percent difference between the experiments error metrics and that of the baseline is given. As it can be seen from Table A.4 the lowest error metrics are obtained for the model trained only on the baseline features. In general addition of the statistical features, the lagged features or both simultaneously increases the prediction errors and decreases the coefficient of determination. As a result of this it can be concluded that the regulation fields at time t contain all the necessary information for making a prediction at least 1 hour later.

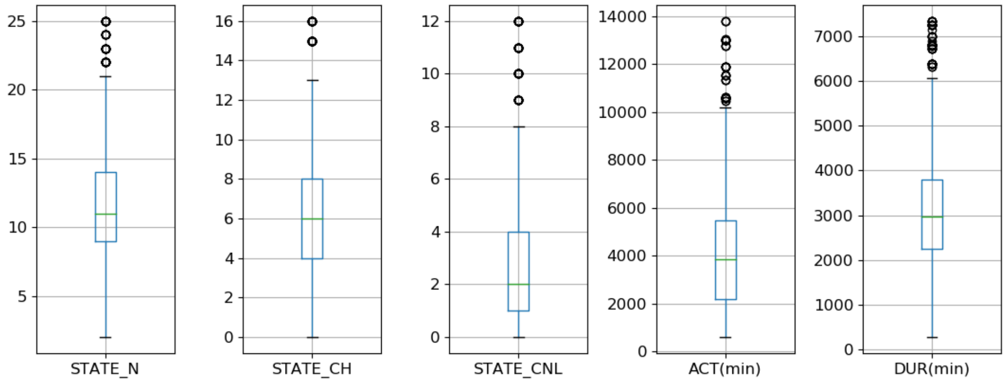


Figure 13: Box and whiskers plot of the values that are to be predicted from the model.

Baseline features											
	STATE_N	STATE_CH	STATE_CNL	ACT(min)	DUR(min)						
RMSE	0.7333	0.5170	0.4497	168.3672	157.4850						
MAE	0.4213	0.2838	0.2396	82.1432	77.0808						
R2	0.9692	0.9677	0.9675	0.9941	0.9834						
Baseline + statistical features						**Percent change compared to baseline**					
	STATE_N	STATE_CH	STATE_CNL	ACT(min)	DUR(min)		STATE_N	STATE_CH	STATE_CNL	ACT(min)	DUR(min)
RMSE	0.7708	0.5272	0.4812	168.8788	159.3960	RMSE	5.12%	1.98%	7.01%	0.30%	1.21%
MAE	0.4627	0.3039	0.2724	85.4782	79.7869	MAE	9.82%	7.11%	13.69%	4.06%	3.51%
R^2	0.9660	0.9665	0.9627	0.9941	0.9830	R^2	-0.33%	-0.13%	-0.49%	0.00%	-0.04%
Baseline + lagged features						**Percent change compared to baseline**					
	STATE_N	STATE_CH	STATE_CNL	ACT(min)	DUR(min)		STATE_N	STATE_CH	STATE_CNL	ACT(min)	DUR(min)
RMSE	0.7988	0.5648	0.4895	175.6788	162.9760	RMSE	8.94%	9.25%	8.86%	4.34%	3.49%
MAE	0.4629	0.3152	0.2728	86.3362	80.7517	MAE	9.86%	11.09%	13.86%	5.10%	4.76%
R^2	0.9635	0.9615	0.9614	0.9936	0.9822	R^2	-0.59%	-0.65%	-0.62%	-0.05%	-0.12%
Baseline + statistical + lagged features						**Percent change compared to baseline**					
	STATE_N	STATE_CH	STATE_CNL	ACT(min)	DUR(min)		STATE_N	STATE_CH	STATE_CNL	ACT(min)	DUR(min)
RMSE	0.8030	0.5426	0.5076	173.0005	160.8744	RMSE	9.51%	4.95%	12.88%	2.75%	2.15%
MAE	0.4740	0.3183	0.2823	85.5241	79.6207	MAE	12.50%	12.17%	17.79%	4.12%	3.30%
R^2	0.9631	0.9645	0.9585	0.9938	0.9827	R^2	-0.63%	-0.34%	-0.92%	-0.03%	-0.07%

Table 1: Results of the feature selection experiments. On the top the baseline results are given. For each of the experiments the error metrics of the test set are shown together with the percent change compared to the baseline.

6.2. Tuning results

Based on the results of the previous section, the 40 baseline features are selected and the hyper-parameters of the random forest are tuned. The tuning procedure consisted of two stages as it has been discussed in subsection 5.6.

First stage tuning results. By varying one hyper-parameter at a time while leaving the others at their default values, the hyper-parameters that would be needed to be tuned where identified. The hyper-parameters that where selected through this procedure consisted of: the number of estimators (number of trees), maximum depth of the tree and the maximum amount of features to sample from when selecting the splitting criteria.

Second stage tuning results. After obtaining the parameters to be tuned from the first stage the parameter grid was constructed by setting number of estimators $= [10, 50, 150, 200, 250, 300]$; maximum depth $= [10, 20, 30, 40]$ and max features $= ['sqrt', 'log2', 1/3, 1/4, 1/5, 1/6, 1/7, 1/8, 1/10]$. An important parameter in the grid search procedure is the scoring function that has to be minimized. As such the grid search was performed two times, the first time using MAE as the scoring function and the second time using MSE. For each time the parameter combination that resulted in the lowest cross-validation error were selected. The found parameter for both cases are given in Table 2. Both these parameter combinations were tested and the results are presented in Table B.5. From this table it can be seen that the parameter combination that brought the biggest reduction of the prediction error consists of the parameters that were found to minimize the mean absolute error scoring function.

	MAE minimized	**MSE minimized**
No. estimators	250	300
Max depth	30	20
Max features	1/4	1/3

Table 2: Best parameter combinations found from the grid search

Since for the splitting criteria for each tree only a subset of the feature space, namely 1/4 of it, will be used it was decided to check the performance of bootstrapping of the sample space. It was found that the test set errors reduced when setting bootstrapping to False (constructing each tree from the full training samples) a comparison of the models with and without bootstrapping is given in Table B.6. The cross-validation errors of the final model are shown in Table 3.

Baseline features, MAE minimized, bootstrapping=False KFold(K=10)					
	STATE_N	STATE_CH	STATE_CNL	ACT(min)	DUR(min)
RMSE	0.5791	0.3752	0.2919	156.6307	115.5600
MAE	0.2743	0.1547	0.1057	61.3711	48.4627
R2	0.9804	0.9854	0.9865	0.9953	0.9913

Table 3: Results of applying KFold validation on the tuned model

6.3. Final model results

After the optimal parameters were found and validated the model was run on this parameters and evaluated on the test set (353 samples). In Figure 15 and Figure 14 the predicted versus the actual values for total duration and total activation time of the list of regulations at $t + \Delta t$ are given, together with the optimal prediction line. As it can be seen the majority of samples are clustered around the optimal line. Looking at the ranges of the actual values it can be seen that a small deviation from the optimal line will lead to high values for the error metrics.

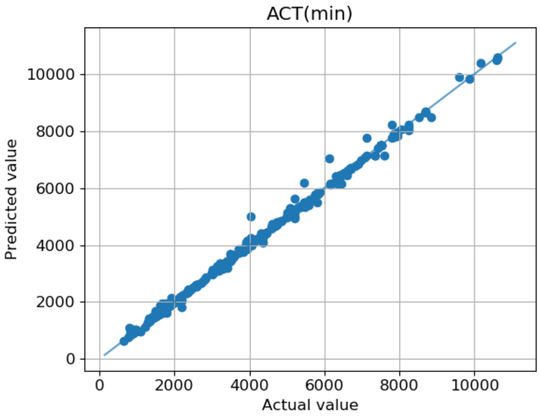


Figure 14: Predicted vs actual total activation time for the list of regulations at $t + \Delta t$.

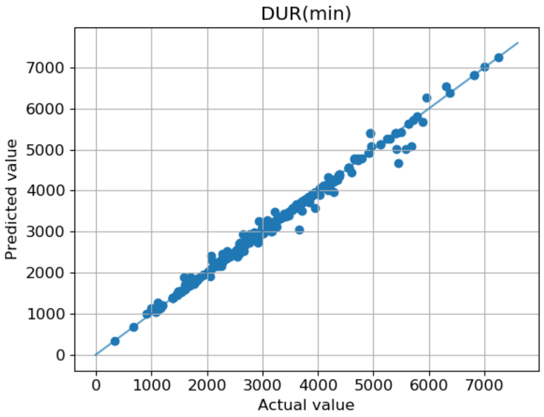


Figure 15: Predicted vs actual total duration of the list of regulations at $t + \Delta t$.

In Figure 16, Figure 17 and Figure 18 the predicted versus actual plot for the total number of new, changed and cancelled regulations at $t + \Delta t$ are given. As it can be seen from these plots and comparing them to the previous two it can be noticed that the predicted values have higher dispersion around the optimal line. This indicates that the problem of predicting these variables is a more difficult problem than the total activation time and duration.

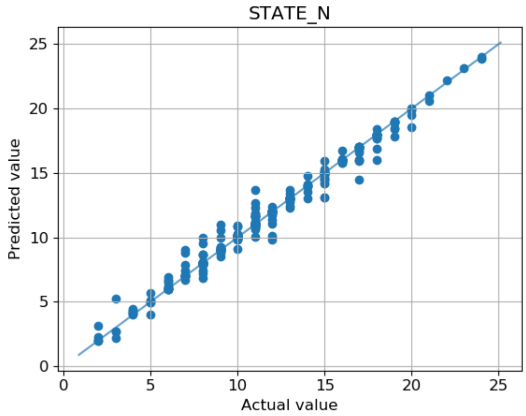


Figure 16: Predicted vs actual total number of new regulations for the list of regulations at $t + \Delta t$.

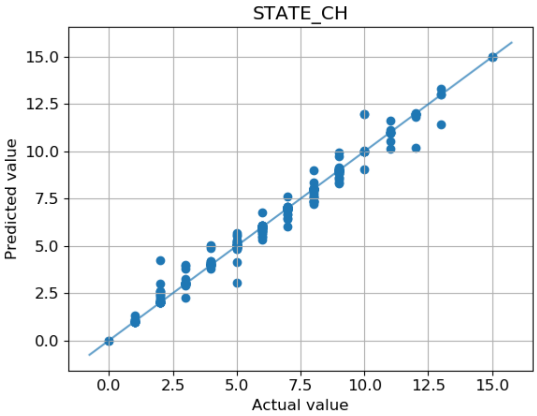


Figure 17: Predicted vs actual total number of changed regulations for the list of regulations at $t + \Delta t$.

In Figure 19 the importance of each feature determined by the Scikit-learn implementation of the Random Forest is given. As it can be seen the top most important features are total activation time, duration and number of pre-tactical regulations from the list of regulations at time t. Activation time is an indicator of the traffic complexity and the way the decisions are being made. At the same time a decreasing number of pre-tactical regulation further indicates changing traffic conditions and inability of the planned regulations to cope with the traffic dynamics. Another interesting observation looking at the status related time variables is that the total time to start of not started regulations is ranked higher than its counterparts for ongoing or finished regulations. Finally the last 4 features with zero importance are due to the fact that in the dataset and ACC they never occur. EDUU is an UAC, thus it handles en-route traffic flying at high altitudes.

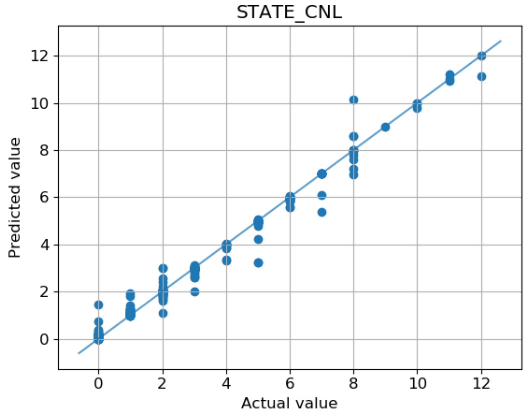


Figure 18: Predicted vs actual total number of cancelled regulations for the list of regulations at $t + \Delta t$.

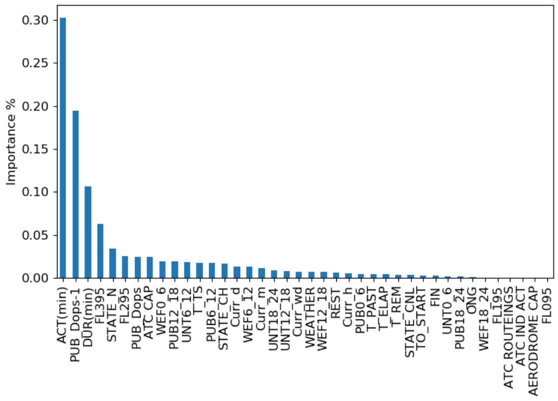


Figure 19: Feature importance determined by the Random Forest model. Sorted from highest importance to lowest.

6.4. Validation of the maximum prediction horizon

After finalization of the model by selecting as input features the baseline features given in Table A.4, the hyper-parameters that minimized MAE shown in Table 2 and setting bootstrapping to False the performance of the model was checked under increasing prediction horizon Δt. As it has been discussed in subsection 5.7, the full dataset of 10 minute snapshots collected over the months of April, May, June, July 2019 was used to create input-output pairs of varying Δt. For each Δt the cross-validation error obtained as the mean error metrics over K training-testing instances of the model, with $K = 10$, were recorded.

In Figure 20 and Figure 21, the RMSE for the states of future regulations and RMSE for total activation time and duration of future regulations respectively are given. In Figure 20 when increasing Δt the RMSE in predicting the number of new regulations at $t + \Delta t$ initially decreases only to start increasing again after $\Delta t = 6$ hours. For predicting the number of new changed regulations at $t + \Delta t$, the RMSE seems to be about constant up to $\Delta t = 6$ hours after which it is increasing. Finally for predicting the number of cancelled regulations at $t + \Delta t$, the value of RMSE is increasing after $\Delta t = 4$ hours. In Figure 21 it can be seen that predicting the total activation time and duration of regulations at $t + \Delta t$, the prediction for these variables start to degrade after $\Delta t = 6$ hours

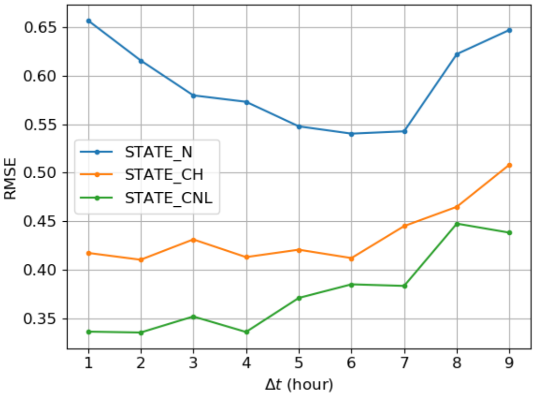


Figure 20: RMSE cross-validation error for the number of new, change and cancelled regulations target predictions as a function of prediction horizon Δt.

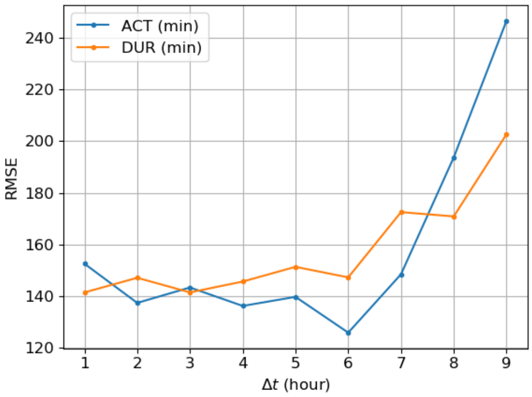


Figure 21: RMSE cross-validation error for the total duration and activation target predictions as a function of prediction horizon Δt.

In Figure 22 the MAE for the future regulation state variables is given. The behaviour of MAE for these variables under increasing Δt seems to behave much like the RMSE for these variables. On the contrary the MAE for the total activation time and duration, has an initial decrease only to increase again after $\Delta t = 5$ hours for total duration and after $\Delta t = 6$ hours for total activation time.

Finally, in Figure 24 and Figure 25 the behaviour of the prediction accuracy as a function of Δt is given. For total activation time and duration there is a clear drop in accuracy after $\Delta t = 6$ hours. For the accuracy of predicting the future number of changed regulations the prediction accuracy seems to be in line with the error metrics, degrading after $\Delta t = 6$ hours. The accuracy of predicting number of new regulations seems to be inline with the behaviour of RMSE and MAE for this variable, initially increasing in accuracy only to drop after $\Delta t = 6$ hours. For number of cancelled regulations the accuracy drops after $\Delta t = 4$ hours much like RMSE and MAE.

From these results it can be seen that the prediction horizon above which the predictions start to degrade is not the same for all target variables. For predicting the the number of cancelled regulations in the future the maximum prediction horizon consistent with all performance metrics used seems to be around 4 hours. For the other two regulation state counting variables this occurs at 6 hours. The same prediction horizon of 6 hours is found for the total activation time and duration. These results suggest that the true prediction horizon for EDUU in the months from April to July 2019 is between 4 and 6 hours.

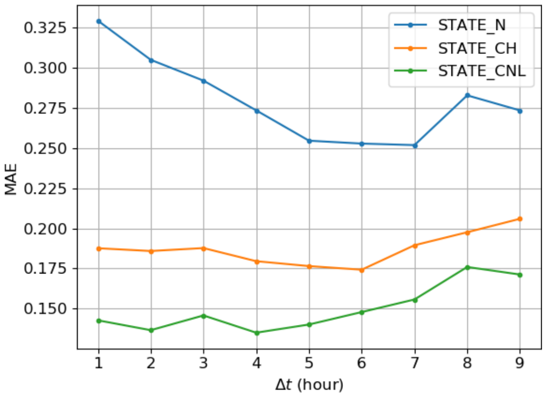


Figure 22: MAE cross-validation error for the number of new, change and cancelled regulations target predictions as a function of prediction horizon Δt.

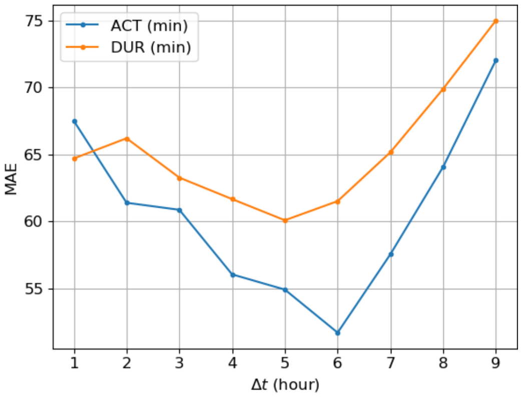


Figure 23: MAE cross-validation error for the total duration and activation target predictions as a function of prediction horizon Δt.

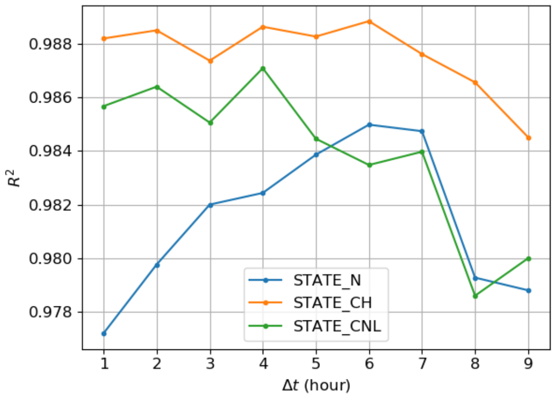


Figure 24: Cross-validation coefficient of determination for the number of new, change and cancelled regulations target predictions as a function of prediction horizon Δt.

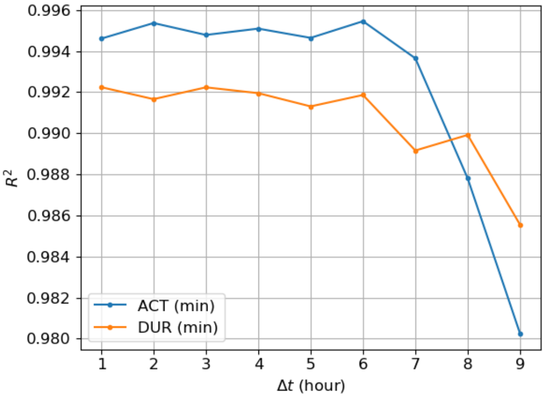


Figure 25: Cross-validation coefficient of determination for the total duration and activation target predictions as a function of prediction horizon Δt.

7. Discussion

In this study there were two primary objectives. The first objective was to create a predictive model with which the evolution of ATFCM regulations could be predicted for different forecast horizons. Utilizing a Random Forest regression model, with the set of features discussed in subsection 6.1 and the hyper-parameters given in subsection 6.2, the number of new, changed, cancelled regulations, total activation time and duration of future regulations were predicted with acceptable error rates.

The largest MAE for the state related variables was 0.27 for number of new regulations, while the smallest possible actual value for these variables is 0. For the total activation time and duration of future regulations the MAE was respectively 61.37 and 48.46 minutes, while the minimum actual values in the data for these target predictions were 601 and 280 minutes. Through tuning the model and setting bootstrapping off an average reduction of 47.14% in MAE across all five target predictions is achieved. The increase in performance by turning off bootstrapping was surprising given the fact that the power of random forest comes from sampling over the training and feature spaces to obtain de-correlated trees [31]. Omitting bootstrapping makes the Random Forest similar to an Extra Randomized Trees model. From the literature considering the problem of predicting arrival/departure delays, the lowest error rates where observed in [8] when utilizing Extra Randomized Trees.

When extending the prediction horizon of the model, it was observed that up to a prediction horizon Δt the performance of the model depending on the target prediction and error metric would either fluctuate around a mean value afterwards increasing or initially decrease only to go up again after Δt. This last behaviour is unexpected, where in the case predicting number new regulations increasing the prediction horizon from 1 hour up to 6 hours led to a 20% decrease of MAE. When the same methodology was applied to LECM and LFEE ACCs this type of behaviour was not observed for any of the five target predictions. This suggests that this outcome may be due to the planning process and ATFCM related decisions in EDUU.

The second objective was to research and implement a methodology with which the maximum prediction horizon for different ACCs/UACs could be estimated. The proposed methodology is based on the Hurst exponent as an indicator of the predictability of time series of ATFCM regulations in the area of consideration. In order to estimate the value for the Hurst exponent the rescaled range method is used as described in subsection 3.1. Through the experiments described in subsection 3.2 and the results of these experiments in subsection 4.1 it was found that:

1. Using the rescaled range method, the estimated Hurst exponent of randomly generated series with uncorrelated increments can deviate from its theoretical value of 0.5 as shown in Figure 6.
2. The rescaled range method was not able to estimate the correct Hurst exponent when the input time series is non-stationary, as indicated from the results in Figure 7 for fractional Brownian motion time series
3. The rescaled range method can not accurately calculate the Hurst exponent of fractional Gaussian noise series. However, it is able to provide a reasonable estimate whether for the input time series the process its increments are negatively/positively correlated or uncorrelated.

With regards to the first finding, for a series length of 1024 samples similar results have been obtained by Qian and Rasheed in [18]. In relation to the second finding, it was observed from previous authors using the rescaled range approach for estimating the Hurst exponent on financial time series [18, 15] is applied on the time series of returns rather than time series of the price. If we consider the price time series $\{p_t\}$, the return time series is obtained as $r_t = p_t - p_{t-1}$ [15]. As shown in [33], this differencing operation is a common tool to convert a non-stationary series into a stationary one. Finally for the third finding, similar results have been obtained in [13] for fGn series with $0.5 \leq H \leq 1$.

By using the TSLHE procedure proposed in [14], the average Hurst exponent as a function of time-scale is obtained. The research hypothesis was that the prediction horizon can be obtained as the time-scale at which the process starts to be random, TS_H. By comparing the monthly evolution of TS_H with the mean return interval of regulations, Figure 9, it was observed that the monthly changes in TS_H seem to match the changes in the rate of occurrence of regulations. At the same time, it was observed that in the case of EDUU and LFMM that for the month of June 2019 a value for TS_H could not be determined potentially indicating the presence of a fully random regime. Based on the findings an inverse relationship is suggested between the frequency of occurrence of regulations and the value of the maximum prediction horizon. This results seem to follow the intuition that the more active a system is, it is more beneficial to focus on the short term predictions and vice-versa.

Verification of proposed method for the maximum prediction horizon TS_H, with the prediction horizon determined from the inverse of the largest Lyapunov exponent T_L, resulted in very similar values. For the case of EDUU it was found $TS_H = 4.77$ and $T_L = 4.62$ hours. TS_H aims to find the time-scale at which the process increments become uncorrelated and random, meanwhile T_L aims to find the time at which a system transitions from deterministic to chaotic and unpredictable. As such it can be concluded that using the time-scale at which the Hurst exponent indicates lack of correlations, is a valid approach for estimating the prediction horizon.

Comparing these values for TS_H and T_L from the value of Δt above which prediction errors start increasing, which for EDUU was found to be between 4 and 6 hours, it can be seen that both methodologies are within this range. The process of obtaining T_L involves several in between steps such as estimating the embedding dimension and selecting the lag for phase space reconstruction, which require different assumptions in selecting these parameters. Furthermore the linear regression region for obtaining the value of LLE is a subjective choice, based on what looks linear. As such it can be the case that the same methodology applied on the same series by two different people may result in two different prediction horizons T_L. In comparison the method with which TS_H is obtained is a simpler one, requiring only basic knowledge of statistics and time series analysis and minimal assumptions.

Finally, in relation to value for TS_H, this value should not be considered as a cut-off point after which predictions should not be made. Instead it should be thought as the point in the future after which a trade-off between prediction accuracy and ability to predict further ahead should be made. As such TS_H is only an estimator of the prediction horizon, the true maximum prediction horizon should be selected based on the needs of the forecast and the acceptable level of accuracy in the predictions.

8. Conclusions

In this research a model with which characteristics of ATFCM regulations, which relate traffic complexity with the performance of currently planned regulations, are predicted and a methodology with which the maximum prediction horizon can be estimated has been proposed. After pre-processing the publicly available ATFCM notification messages, it was able to train a Random Forest given the list of ANMs at time t to predict the number of new, changed, cancelled regulations, total activation time and duration of the regulations at $t + \Delta t$, where Δt is the prediction horizon. In the case of predicting the future regulations over Karlsruhe Upper Area Control, it was found that the prediction errors of the aforementioned targets are stable up to a prediction horizon between 4 and 6 hours after which they start increasing. Given the fact that the data utilized are publicly available the proposed model offers the opportunity to make predictions on measures used for demand-capacity balancing without needing data on actual capacities, current demand and future demand.

Utilizing the Time-Scale Local Hurst Exponent [14], it was able to determine the time-scale at which the process under study starts becoming random, TS_H. Comparing this time-scale with the prediction horizon obtained from the inverse of the largest Lyapunov exponent [23], it was concluded that the proposed methodology is a valid estimator of the prediction horizon. The methodology was further validated by extending the prediction horizon of the predictive model. It was found that the proposed method is able to give an estimate of the prediction horizon above which the prediction errors start to increase. As such the proposed methodology can be utilized to estimate the point in time in the future after which a trade off between predictive accuracy and ability to predict further ahead should be made. In this research the input time series for which the prediction horizon was estimated consisted of integer time series, much like demand time series in the fields of manufacturing and maintenance. As a result of this, the proposed method could potentially be used as a decision support tool in industries that rely on forecasts for their operations, to determine the prediction horizon based on the operational needs and acceptable level of errors in predictions.

9. Future work

The advantage of the proposed predictive model in this study is that the data utilized to make predictions on the future ATFCM regulations are publicly available. However, at the same time the data utilized only reflect the outcome of the process that is to be predicted and does not contain information about the causal factors that lead to changes in the regulatory situation. As such the first point of improvement would be inclusion of data such as weather information, flight schedules/trajectories and capacity information so that a causal understanding of the factors that lead to new, changed and cancelled regulations can be obtained.

For most stakeholders in air transportation it also important to know before hand what the impact of the planned measures will be in terms of ATFM delay. Thus, the second point of improvement is to utilize the predictions made by the proposed model as inputs to a second model with which the future impact of regulations can be predicted. So far the spatial scope of the predicted regulations is only defined by the geographical location of the ACC/UAC and no consideration is given in predicting which flight levels will be regulated. Further refinement of the geographic scope to elementary sectors within the ACC/UAC is of limited advantage as these sectors are subject to changes under ATFCM measures for capacity optimization. However substantial value can be obtained by being able to predict within the ACC/UAC the flight levels that will be regulated.

As a practical application of the proposed model consider an aircraft operator, scheduled for an IFR flight. At least three hours from the planned departure time the aircraft operator has the ability to modify its flight plan. If reliable predictions on which ACCs/UACs and which flight levels within them will be regulated and the expected impact of these regulations for different prediction horizons are available, the aircraft operator can leverage these predictions in order to modify the original flight plan. Given the above predictions the flight plan can be posed as an optimization problem with the objective of finding the trajectory that minimizes ATFM delay on the flight. In this way the operator can reduce operational costs and travel time for its passengers. From the perspective of the network this optimized trajectory will move this flight into an area or flight level where a regulation is not expected, therefore offloading the imbalance on the regulated sectors.

References

[1] EUROCONTROL Statistics and Forecast Service, European aviation in 2040 - Challenges of growth (2018).

[2] Central Office for Delay Analysis, CODA DIGEST 2018, Tech. rep., EUROCONTROL (2019).

[3] G. Lenti, Understanding a difficult Summer 2018: why it happened? (Jan 2019).
URL `https://www.eurocontrol.int/nm-user-forum-2019`

[4] S. Niarchakou, M. Cech, ATFCM Operations Manual, 23rd Edition, EUROCONTROL, Brussels, Belgium, 2019.

[5] Y. Liu, M. Hansen, Predicting the initiation of a ground delay program, Journal of Aerospace Operations 5 (1) (2018) 75–84.

[6] A. Estes, M. O. Ball, D. J. Lovell, Predicting performance of ground delay programs, Twelfth USA/Europe Air Traffic Management Research and Development Seminar.

[7] K. Shetty, J. Gulding, H. Koelman, M. Celiktin, R. Koelle, Comparison of ATFM practices and performance in the US and Europe, in: 2017 Integrated Communications, Navigation and Surveillance Conference (ICNS), IEEE, 2017, pp. 1C1–1.

[8] B. Thiagarajan, L. Srinivasan, A. V. Sharma, D. Sreekanthan, V. Vijayaraghavan, A machine learning approach for prediction of on-time performance of flights, 2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC).

[9] S. Manna, S. Biswas, R. Kundu, S. Rakshit, P. Gupta, S. Barman, A statistical approach to predict flight delay using gradient boosted decision tree, 2017 International Conference on Computational Intelligence in Data Science(ICCIDS).

[10] A. Bunde, J. F. Eichner, S. Havlin, J. W. Kantelhardt, Return intervals of rare events in records with long-term persistence, Physica A: Statistical Mechanics and its Applications 342 (2004) 308–314. `doi:10.1016/s0378-4371(04)00487-x`.

[11] B. B. Mandelbrot, J. W. Van Ness, Fractional brownian motions, fractional noises and applications, SIAM review 10 (4) (1968) 422–437.

[12] Q. Yuan, W. Zhou, S. Li, D. Cai, Epileptic EEG classification based on extreme learning machine and nonlinear features, Epilepsy research 96 (1-2) (2011) 29–38.

[13] T. Karagiannis, M. Molle, M. Faloutsos, Long range dependence- ten years of internet traffic modelling, IEEE Internet Computing 8 (5) (2004) 57–64. `doi:10.1109/mic.2004.46`.

[14] E. Molino-Minero-Re, F. García-Nocetti, H. Benítez-Pérez, Application of a time-scale local hurst exponent analysis to time series, Digital Signal Processing 37 (2015) 92–99.

[15] M. Raimundo, J. Okamoto Jr, Application of Hurst Exponent (H) and the R/S analysis in the classification of FOREX securities, International Journal of Modeling and Optimization 8 (2018) 116–124. doi:10.7763/IJMO.2018.V8.635.

[16] C. Wang, Z. Zhang, M. Zhu, Nonlinear dynamic analysis of air traffic flow at different temporal scales: Nonlinear analysis approach versus complex networks approach, 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)doi:10.1109/qrs-c.2018.00079.

[17] B. B. Mandelbrot, J. R. Wallis, Robustness of the rescaled range R/S in the measurement of noncyclic long run statistical dependence, Water Resources Research 5 (5) (1969) 967–988.

[18] B. Qian, K. Rasheed, Hurst exponent and financial market predictability, in: IASTED conference on Financial Engineering and Applications, 2004, pp. 203–209.

[19] S. Niarchakou, ATFCM Users Manual, 23rd Edition, EUROCONTROL, Brussels, Belgium, 2019.

[20] C. Flynn, Python package for fractional Brownian motion and fractional Gaussian noise (2019).
URL https://github.com/crflynn/fbm

[21] R. B. Davies, D. Harte, Tests for hurst effect, Biometrika 74 (1) (1987) 95–101.

[22] T. Dieker, Simulation of fractional brownian motion, Master's thesis, Department of Mathematical Sciences, University of Twente (2004).

[23] S. H. Strogatz, Nonlinear dynamics and chaos: with applications to physics, biology, chemistry, and engineering, CRC Press, 2018.

[24] M. Alfaro, G. Fuertes, M. Vargas, J. Sepúlveda, M. Veloso-Poblete, Forecast of chaotic series in a horizon superior to the inverse of the maximum lyapunov exponent, Complexity 2018.

[25] M. T. Rosenstein, J. J. Collins, C. J. De Luca, A practical method for calculating largest Lyapunov exponents from small data sets, Physica D: Nonlinear Phenomena 65 (1-2) (1993) 117–134.

[26] F. Takens, Detecting strange attractors in turbulence, in: Dynamical systems and turbulence, Warwick 1980, Springer, 1981, pp. 366–381.

[27] L. A. Aguirre, C. Letellier, Modeling nonlinear dynamics and chaos: a review, Mathematical Problems in Engineering 2009.

[28] A. M. Fraser, H. L. Swinney, Independent coordinates for strange attractors from mutual information, Phys. Rev. A 33 (1986) 1134–1140. doi:10.1103/PhysRevA.33.1134.
URL https://link.aps.org/doi/10.1103/PhysRevA.33.1134

[29] M. B. Kennel, R. Brown, H. D. Abarbanel, Determining embedding dimension for phase-space reconstruction using a geometrical construction, Physical review A 45 (6) (1992) 3403.

[30] A. Géron, Hands-On Machine Learning with Scikit-Learn & TensorFlow, O'Reilly Media, 2017.

[31] J. Friedman, T. Hastie, R. Tibshirani, The elements of statistical learning, Vol. 1, Springer series in statistics New York, 2001.

[32] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[33] R. J. Hyndman, G. Athanasopoulos, Forecasting: principles and practice, OTexts, 2018.

A. Features considered in feature selection

	Baseline features	Description	Statistical features	Description	Lagged features	Description
1	WEF0_6	No. regs. starting between 0 and 6	max_T_PAST	Max of non-zero T_PAST$_i$	$L_1\Delta$PUB$_{Dops}$	Change in no. regs published during Dops (1h ago)
2	WEF6_12	No. regs. starting between 6 and 12	min_T_PAST	Min of non-zero T_PAST$_i$	$L_1\Delta$PUB$_{Dops-1}$	Change in no. regs published during Dops-1 (1h ago)
3	WEF12_18	No. regs. starting between 12 and 18	avg_T_PAST	Mean of non-zero T_PAST$_i$	$L_1\Delta$FIN	Change in no. finished regs. (1h ago)
4	WEF18_24	No. regs. starting between 18 and 24	std_T_PAST	Stand. dev. of non-zero T_PAST$_i$	$L_1\Delta$ONG	Change in no. ongoing refs (1h ago)
5	UNT0_6	" " finishing " " "	max_T_ELAP	Max of non-zero T_ELAP$_i$	$L_1\Delta$TO_START	Change in no. of regs to start (1h ago)
6	UNT6_12	" " finishing " " "	min_T_ELAP	Min of non-zero T_ELAP$_i$	$L_1\Delta$STATE_N	Change in no. of new regs. (1h ago)
7	UNT12_18	" " finishing " " "	avg_T_ELAP	Mean of non-zero T_ELAP$_i$	$L_1\Delta$STATE_CH	Change in no of changed regs. (1h ago)
8	UNT18_24	" " finishing " " "	std_T_ELAP	Stand. dev. of non-zero T_ELAP$_i$	$L_1\Delta$STATE_CNL	Change in no of cancelled regs. (1h ago)
9	PUB0_6	" " published " " "	max_T_REM	Max of non-zero T_REM$_i$	$L_1\Delta$ACT	Change in total activation time (1h ago)
10	PUB6_12	" " published " " "	min_T_REM	Min of non-zero T_REM$_i$	$L_1\Delta$DUR	Change in total duration (1h ago)
11	PUB12_18	" " published " " "	avg_T_REM	Mean of non-zero T_REM$_i$	$L_1\Delta$ATC_CAP	Change in no. of ATC capacity regs. (1h ago)
12	PUB18_24	" " published " " "	std_T_REM	Stand. dev. of non-zero T_REM$_i$	$L_1\Delta$AEROD_CAP	Change in no. of aerodrome capacity regs. (1h ago)
13	PUBDops	No. regs. published on Dops	max_T_TS	Max of non-zero T_TS$_i$	$L_1\Delta$ATC_IND_ACT	Change in no. of regs. from atc strikes (1h ago)
14	PUBDops-1	No. regs. published on Dops -1	min_T_TS	Min of non-zero T_TS$_i$	$L_1\Delta$ATC_ROUTEING	Change in no . ATC reouteings regs. (1h ago)
15	STATE_N	No. of new regs.	avg_T_TS	Mean of non-zero T_TS$_i$	$L_1\Delta$WEATHER	Change in no. of weather regs. (1h ago)
16	STATE_CH	No. of changed regs	std_T_TS	Stand. dev. of non-zero T_TS$_i$	$L_1\Delta$REST	Change in no. of rest regs. (1h ago)
17	STATE_CNL	No. of cancelled regs	n_neg_ACT	No. of negtive activation times	$L_2\Delta$PUB$_{Dops}$	Change in no. regs published during Dops (2h ago)
18	ACT	Total activation time of regs. (minutes)	min_ACT	Minimum of non-zero act. times	$L_2\Delta$PUB$_{Dops-1}$	Change in no. regs published during Dops-1 (2h ago)
19	DUR	Total duration of regs. (minutes)	min_pos_ACT	Minimum of positive act. times	$L_2\Delta$FIN	Change in no. finished regs. (2h ago)
20	FL395	No. regs. blocking FL395 to FL295	avg_ACT	Mean of non-zero act. times	$L_2\Delta$ONG	Change in no. ongoing refs (2h ago)
21	FL295	No. regs. blocking FL295 to FL195	std_ACT	Stand dev. of non-zero act. times	$L_2\Delta$TO_START	Change in no. of regs to start (2h ago)
22	FL195	No. regs. blocking FL195 to FL095	max_DUR	Maximum of non-zero durations	$L_2\Delta$STATE_N	Change in no. of new regs. (2h ago)
23	FL095	No. regs. blocking FL095 to FL000	min_DUR	Minimum of non-zero durations	$L_2\Delta$STATE_CH	Change in no of changed regs. (2h ago)
24	ATC_CAPACITY	No. of regs. due to ATC capacity	avg_DUR	Mean of non-zero durations	$L_2\Delta$STATE_CNL	Change in no of cancelled regs. (2h ago)
25	AERODROME_CAPACITY	No. of regs. due to aerodrome capacity	std_DUR	Stand. dev. of non-zero durations	$L_2\Delta$ACT	Change in total activation time (2h ago)
26	ATC_IND_ACTION	No. of regs. due to ATC strikes			$L_2\Delta$DUR	Change in total duration (2h ago)
27	ATC_ROUTEINGS	No. of regs. due to ATC routeing			$L_2\Delta$ATC_CAP	Change in no. of ATC capacity regs. (2h ago)
28	WEATHER	No. of regs. due to weather			$L_2\Delta$AEROD_CAP	Change in no. of aerodrome capacity regs. (2h ago)
29	REST	No. of regs due to remaining reasons			$L_2\Delta$ATC_IND_ACT	Change in no. of regs. from atc strikes (2h ago)
30	FIN	No. of regs that have finished			$L_2\Delta$ATC_ROUTEING	Change in no . ATC reouteings regs. (2h ago)
31	ONG	No. of regs that are ongoing			$L_2\Delta$WEATHER	Change in no. of weather regs. (2h ago)
32	TO_START	No. of regs that have not started yet			$L_2\Delta$REST	Change in no. of rest regs. (2h ago)
33	T_PAST	Total time past for finished regs.				
34	T_ELAP	Total time elapsed for ongoing regs.				
35	T_REM	Total time remaining for ongoing regs.				
36	T_TS	Total time to start for not started regs.				
37	Curr_m	Current month of the year (1-12)				
38	Curr_d	Current day of the month (1-31)				
39	Curr_wd	Current day of the week (0:Mon - 6:Sun)				
40	Curr_h	Current hour of the day (decimal hour)				

Table A.4: Overview of the input feature and short description for each of them

B. Second stage tuning results

Baseline features

	STATE_N	STATE_CH	STATE_CNL	ACT(min)	DUR(min)
RMSE	0.7333	0.5170	0.4497	168.3672	157.4850
MAE	0.4213	0.2838	0.2396	82.1432	77.0808
R2	0.9692	0.9677	0.9675	0.9941	0.9834

Baseline features MAE minimized						**Percent change compared to baseline**					
	STATE_N	STATE_CH	STATE_CNL	ACT(min)	DUR(min)		STATE_N	STATE_CH	STATE_CNL	ACT(min)	DUR(min)
RMSE	0.5940	0.4262	0.3574	148.1839	126.8990	RMSE	-18.99%	-17.56%	-20.53%	-11.99%	-19.42%
MAE	0.3481	0.2298	0.1839	75.6494	67.5146	MAE	-17.37%	-19.00%	-23.25%	-7.91%	-12.41%
R2	0.9798	0.9781	0.9794	0.9954	0.9892	R2	1.09%	1.07%	1.24%	0.13%	0.59%
Baseline features MSE minimized						**Percent change compared to baseline**					
	STATE_N	STATE_CH	STATE_CNL	ACT(min)	DUR(min)		STATE_N	STATE_CH	STATE_CNL	ACT(min)	DUR(min)
RMSE	0.6217	0.4278	0.3776	154.3018	132.1854	RMSE	-15.22%	-17.25%	-16.03%	-8.35%	-16.06%
MAE	0.3623	0.2313	0.1915	74.9772	68.6698	MAE	-13.99%	-18.48%	-20.07%	-8.72%	-10.91%
R2	0.9779	0.9779	0.9771	0.9951	0.9883	R2	0.89%	1.05%	0.99%	0.09%	0.50%

Table B.5: Test set errors after constructing the models using the optimal parameters from the grid search.

Baseline features, MAE minimized, bootstrapping = True

	STATE_N	STATE_CH	STATE_CNL	ACT(min)	DUR(min)
RMSE	0.5940	0.4262	0.3574	148.1839	126.8990
MAE	0.3481	0.2298	0.1839	75.6494	67.5146
R2	0.9798	0.9781	0.9794	0.9954	0.9892

Baseline feature, MAE minimized, bootstrapping = False						**Percent change compared to baseline**					
	STATE_N	STATE_CH	STATE_CNL	ACT(min)	DUR(min)		STATE_N	STATE_CH	STATE_CNL	ACT(min)	DUR(min)
RMSE	0.5326	0.3661	0.2965	125.7716	109.5088	RMSE	-10.34%	-14.10%	-17.05%	-15.12%	-13.70%
MAE	0.2542	0.1560	0.1131	53.8917	47.6721	MAE	-26.97%	-32.14%	-38.48%	-28.76%	-29.39%
R2	0.9838	0.9838	0.9859	0.9967	0.9920	R2	0.40%	0.59%	0.65%	0.13%	0.28%

Table B.6: Effect of bootstrapping of the training instances on the prediction errors.

II

Literature study (as graded under AE4020)

1 Introduction

Air transport has never been cheaper and as accessible as it is in the moment[1]. The surge in flights operated by low cost carriers has made it possible to a bigger percentage of the global population to utilize air transportation. Not only a bigger percentage of the population is flying, but it is also flying more frequently. As such it can be expected that in the years to come demand for air travel will continue to increase. According to Eurocontrol's Statistics and Forecast service in the most likely scenario by 2040 an increase of 53% in IFR movements within Europe can be expected [39]. In Figure 1.1, the average daily traffic over the course of 5 years from 2013 to 2018 is given[25]. As it can be seen in this figure there is a clear increasing trend in average daily traffic. In Figure 1.2 the average daily traffic for each month from 2015 to March 2019 is given. The increasing trend seen in Figure 1.1 causes that for each month the average number of flights to be increasing on a year to year basis adding to the load of the existing European air traffic system.

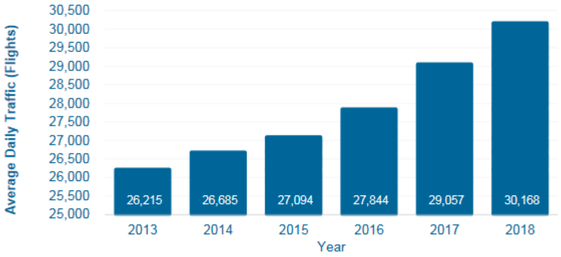


Figure 1.1: Average daily traffic from 2013 to 2018 [25]

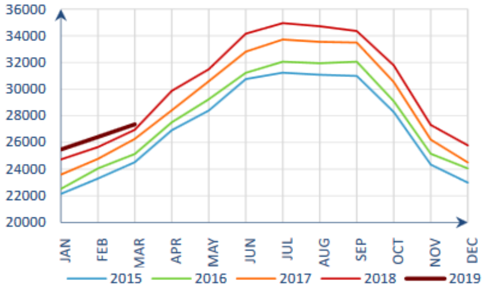


Figure 1.2: Average daily traffic for the last 5 years [43]

One of the biggest effects of this increase in air traffic is an increase of air travel delay. In Figure 1.3, the average departure delay per flight for each month from 2014 to 2018 is given, the average is taken over all delay causes. The situation for arrival delays is similar to departure delays, therefore only the departure statistics are shown. As it can be seen from Figure 1.3, in general the delay peaks for all five years occur during the summer months with some exceptions in December. These summer months are associated with the highest values of average daily traffic as it can be seen in Figure 1.2. Another observation that can be made looking at Figure 1.3 is that the delay curve for the summer of 2018 has increased considerably compared to the previous four years. Summer of 2018 has been regarded from Eurocontrol's Network Manager as a very difficult period for all stakeholder involved in the air traffic system[5]. The actual delays in this period of the year almost doubled when compared to the actual delays of the same period in 2017.

The main reason for the increase in air travel delays during the summer months of 2018 has been reported to be air traffic control (ATC) staffing issues[23]. This lead to a situation where there were more flights than there was capacity to safely control them. Other reasons that contributed to the increased delays include weather conditions and air traffic controller (ATCo) strikes [8]. The European air traffic management (ATM) network is expected to face same order of magnitude delays if not higher in 2019 and this situation will worsen if the ATM system will not be able to match the air traffic growth [23].

ATM is one of the five building blocks that compose the Air Navigation Services (ANS). In it self it is composed of three sub-blocks. The first one is Airspace Management (ASM). ASM is defined as "the planning

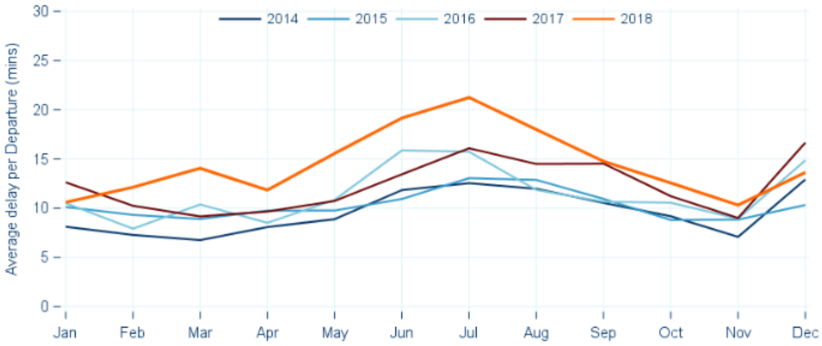


Figure 1.3: Average departure delay per flight, from all causes [8]

function with the objective of maximizing the the utilization of available capacity" [40]. The second block is Air Traffic Services (ATS). ATS in itself is composed of smaller sub-blocks the most notable of which is ATC, whose objectives are "preventing collisions between aircraft or between aircraft and obstacles and expediting and maintaining an orderly flow of traffic" [18]. Finally the third block is composed of Air Traffic Flow Management (ATFM) , which is defined by ICAO as a service whose objective is to contribute to the safety and efficiency of the traffic flows through ensuring that available capacity is fully utilized and the traffic volume is such that that it matches the capacities declared[18]. Both ATFM and ASM are aimed at aiding ATC in meeting its goals. The integration of these blocks in ATM allows for the most efficient usage of airspace and capacity. A representative diagram of the complete ANS is given in Figure 1.4. Note that in Figure 1.4 ATFM is referred to as Air Traffic Flow and Capacity Management (ATFCM). ATFCM a term used by Eurocontrol and the difference between the two terms will be explained in the next sections.

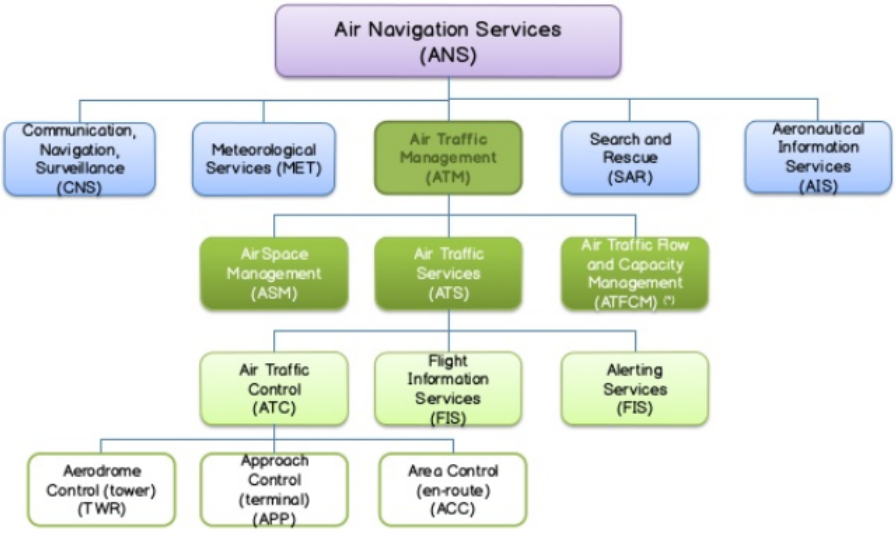

Air Navigation Services (ANS)
Communication, Navigation, Surveillance (CNS)
Meteorological Services (MET)
Air Traffic Management (ATM)
Search and Rescue (SAR)
Aeronautical Information Services (AIS)
Air Space Management (ASM)
Air Traffic Services (ATS)
Air Traffic Flow and Capacity Management (ATFCM) (*)
Air Traffic Control (ATC)
Flight Information Services (FIS)
Alerting Services (FIS)
Aerodrome Control (tower) (TWR)
Approach Control (terminal) (APP)
Area Control (en-route) (ACC)

Figure 1.4: Overall composition of Air Navigation Services [27]

From the above discussion it is clear that within Europe one of the biggest bottlenecks that causes delays is the ATM system and its infrastructure. As such, the purpose of this study is to address this problem and potentially reduce the impact of this bottleneck. Incentivized by the delays of summer 2018, the research will be focused on ATFCM. In particular this study is aimed at bringing forth improvement by utilizing the existing infrastructure and data available. Air transportation is renowned for the amount, type and precision of datasets that have been collected throughout the years. As a result of this, the envisioned outcome of the study is to utilize the available datasets to create a model that will assist decision makers in ATM in the tactical phase of operations by allowing them to make better predictions for the future behaviour of the system.

In the following section background information on the objectives, stakeholders, measures, areas affected and phases of air traffic flow management will be given. This section then will be followed by a discussion of the literature considered for this research. Specifically in chapter 3, the literature related to machine learning modelling for making predictions in air travel is discussed. In chapter 4, the literature read from the field of non-linear analysis is treated. This document is concluded in chapter 5, where the research objectives and questions have been defined together with a preliminary research plan.

2
Air Traffic Flow Management

Taking motivation from the discussion in the previous chapter about the increased delays that occurred during the summer months of 2018, the focus of this study is chosen to be ATFM. As it was mentioned in the previous chapter, the main reason for the increased delays in summer 2018 was due to a situation where the available ATC capacity was not able to safely control all the demand coming from flights. This issue in the aviation industry is referred to as a demand-capacity imbalance. The primary objective of ATFM is to plan and execute measures for demand-capacity balancing (DCB). In the next sections in this chapter, background information on ATFM is going to be presented.

2.1. ATFM Background

As it was mentioned ATFM has two primary objectives. The first one is to optimize the available capacity and the second objective is to protect ATC from excessive demand that can not be handled. Capacity in aviation is defined as the number of flights that can be safely managed in an airspace sector during a time period (typically the hourly rate). Capacities, for the purpose of ATFM are provided by Air Navigation Service Providers (ANSP), based on the available workforce and their experience. Demand can be defined as the number of aircraft that plan to enter an air sector during a given period of time. The demand value is based on the filed flight plans, before the flight takes place.

Several stakeholders are involved in the ATFM process. The main stakeholders are Aircraft Operators(AOs), airports and ANSPs. It can often occur that the objectives of these three stakeholders are not aligned. As a result of this, in order for ATFM to be a successful process the input of each of these stakeholders has to be taken into account. In order to manage this process and to guarantee transparency and fairness to all the actors involved the European Commission created the role of Network Manager (NM) and nominated EUROCONTROL for the role. The responsibilities assigned to the NM by the European Commission include centralized management of the European ATM network and management of rare resources, through a collaborative decision making (CDM) process.

In Figure 1.4, it was mentioned that ATFM is referred to ATFCM. Besides the addition of the "C" in the name for "Capacity", there are significant differences between the two acronyms. The first and most important is that ATFCM can be considered as a part of the ATFM process. While ATFM refers to the collaborative planning and decision making process for ensuring maximum utilization of capacity and safety of traffic flows, ATFCM refers to the measures to be executed in order to achieve the ATFM high level objectives. Another distinction that can be made between the two acronyms is that, while ATFM includes all the 4 main stakeholders (NM, AOs, airports, ANSPs) for ATFCM the relevant stakeholders are the NM and ANSPs. In particular an important role from the side of the ANSPs, is that of the Flow Management Position(FMP) . The FMP aids the NM in its flow management duties as a local expert in a particular area control center(ACC).In Figure 2.1 the difference between the two acronyms is visually described.

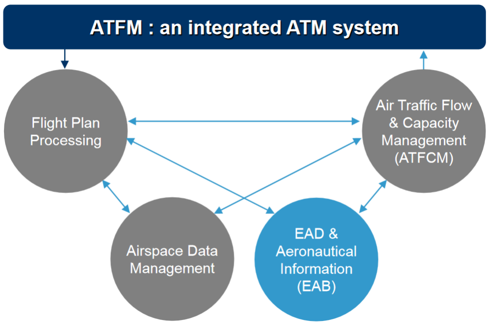

ATFM : an integrated ATM system
Flight Plan Processing
Air Traffic Flow & Capacity Management (ATFCM)
Airspace Data Management
EAD & Aeronautical Information (EAB)

Figure 2.1: Schematic overview of ATFM and constituent processes [7]

2.2. ATFCM Measures

In order to deal with demand-capacity imbalances, the NM together with the affected FMP jointly consider potential solutions and make a decision on the most suitable to be executed. One of the requirements of this process is to take into account the needs of all the stakeholders before a measure is implemented. In the next sections the measures that are considered to resolve demand-capacity issues and their hierarchical order will be stated.

2.2.1. Optimizing the available capacity

The first measures to be considered are related to optimization of the available capacity. This can come in the form of sector management, where the configuration of a particular sector is changed either by splitting it into smaller ones or collapsing smaller sectors to a single one. Another method of optimizing the capacity involves negotiating extra capacity. This can be achieved through implementation of holding patterns, reducing traffic complexity or coordinating with military for airspace usage [30].

2.2.2. Shifting the demand into other areas

Once the class of solutions mentioned in subsection 2.2.1 have been exhausted and there still is a capacity imbalance the Network Managers Operation Center(NMOC) will attempt to shift demand into areas where capacity is still available. This can involve re-routing of traffic flows or flight level management, where part of the flows are assigned to fly at different flight levels. Other solutions of this class involve advancing traffic that is able of departing earlier and tactical interventions of the FMP related to re-routing or change of flight level

2.2.3. Regulating the demand

After the measures mentioned in subsection 2.2.1 and subsection 2.2.2 have failed to solve the imbalance the last option available is to regulate the demand. In order to achieve demand regulation there are two main methods that can be applied. The first one is ground holding, where all the aircraft that meet some criteria are mandated to stay on ground until further notice. This type of measure is usually implemented in cases of severe weather conditions, in cases of accidents or to mitigate long periods of in-flight holding. The duration of this holding period is dependant on the phenomena that caused it, as such the delay induced on affected flights can not be predicted. The other alternative and the focus of this research is application of ATFCM regulations. These regulations are applied for a variety of reasons and in general they are classified into 14 different categories. These different reasons for regulating the demand are given in Figure 2.2, where for each reason the associated total ATFM delay for the month of March 2019 is given.

Once controlled sectors where demand exceeds capacity have been identified, the NM activates ATFCM regulations for these sectors. Flights that will be entering regulated sectors are assigned a pre-departure delay on ground, this delay is referred to as ATFM delay. The process of assigning ATFM delays to individual flights is known as the slot allocation process. For all flights entering the NM area of operations it is mandatory to file a flight plan. These plans are collected in the Integrated Initial Flight Plan Processing System (IFPS), which are then fed together with the ANSPs declared capacities to the Enhanced Tactical Flow Management System (ETFMS). These inputs are then used in the Computer Assisted Slot Allocation (CASA) system.

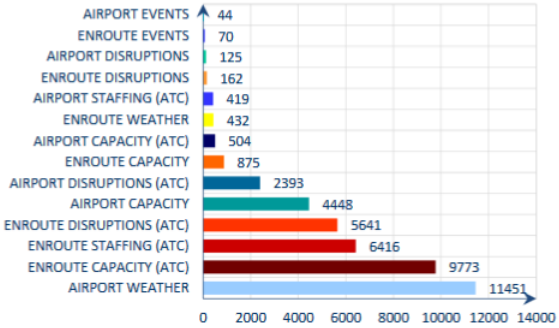


Figure 2.2: Reasons for ATFM delays in March 2019 [43]

To explain the slot allocation process consider a flight departing from *A* and arriving at *B*, whose trajectory passes through 5 controlled sectors. For each sector an estimated time over (ETO) is determined, based on the expected time to enter the sector. Assume that sector three (somewhere in the middle of the trajectory) and sector five (at the destination) have demand capacity imbalances. Slot times are based on the capacities, so if the capacity is 30 flights/hour there is an available slot every two minutes to enter the sector. The first available slots for both regulated sectors are checked and the most penalizing one is chosen. Consider that the ETO for sector three originally was at 17:18 and the first available slot to enter this sector is 17:22, for sector five the original ETO was at 17:51 and the first available slot is 17:57. In this case the slot for entering sector five at 17:57 will be selected as it is most penalizing with a delay of 6 minutes compared to 4 minutes for the other slot. The result of this that the Estimated Take-off Time (ETOT) of the flight will be pushed by 6 minutes, so that it can enter the regulated sector at the defined slot time. Finally when several flights are considered for the process the slot allocation is performed in a ’First Planned, First Served’ principle [30]. This principle can be understood as "Flights should arrive over the regulated area in the same order (based on ETOs) in which they would if there was no regulation".

2.3. ATFM Phases

In order to have an optimal traffic flow over Europe and at the same time take into account the interests of all the affected stakeholders, ATFM is applied on four different time horizons. This involves a planning process for each calendar day, that for the purposes of the ATFM process will be referred to as "Day of Operations" or D_{ops}. In the next paragraphs the different ATFM phases, the activities involved in each of them, the input and outputs for each phase will be stated. Finally in Figure 2.3 the inputs and outputs of each phase are visually illustrated.

Strategic Flow Management This phase occurs seven or more days before D_{ops}. The focus of the phase is on research and planning for identifying major demand-capacity imbalances. All the relevant stakeholders are involved and aim to share information to ensure that the activities are coordinated. The final output of the phase is the Network Operation Plan (NOP)

Pre-tactical Flow Management This takes place six day prior to D_{ops}. Coordination continues and the initial plans are further refined. The focus of the phase is to optimize efficiency and solve imbalances by optimizing available capacity or shifting demand into areas where capacity is available. The output of the phase is the ATFCM Daily Plan(ADP) , which is published through ATFCM Notification Messages(ANM).

Tactical Flow Management Takes place on D_{ops}. Events are considered in real-time and changes to the original plans are made as required. This phase is aimed at ensuring that the measures proposed in the previous two phases are the bare minimum required to solve the imbalance issues. The output of this phase consists of short term forecasts, analysis of impact and maximization of available capacity.

Post-operational Analysis This takes place the day after D_{ops}. The focus is on analysis of the measures used, investigation and reporting on the operational processes. All stakeholders during this phase have the opportunity to present feedback on the efficiency of the ADP. The output of the phase consists of the best practices and actions to be avoided that are fed-back to the process for improvement.

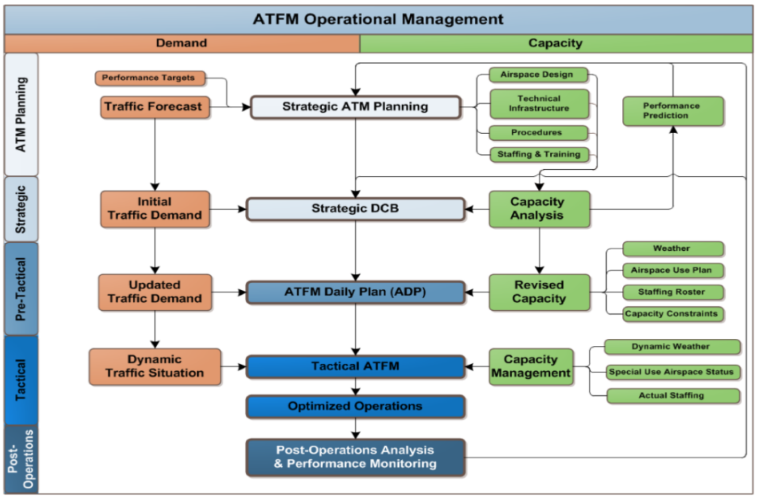
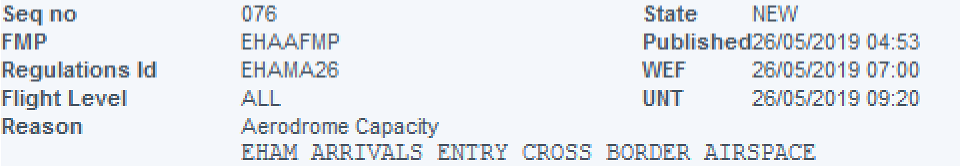


Figure 2.3: ATFM phases diagram [13].

2.4. Characteristics of ATFCM regulations

During the tactical phase of operations, ATFCM regulations that are planned to be activated are communicated to aircraft operators and ANSPs through the ATFCM Notification Messages (ANM). The messages are publicly avaliable and can be accessed by anyone through the Network Operations Portal [1]. A screenshot of such a message is given in Figure 2.4. The message starts of with the *State* indicating whether it is a *NEW*,*CHANGED* or *CANCELED* ATFCM regulation. This is then followed by the relevant FMP, which in the case of Figure 2.4 is the Amsterdam ACC. The FMP name implicitly defines the sector that is being regulated. There are three different times associated with a regulation, the publication time (*Published*), the start time of the regulation (*WEF*) and the stop time (*UNT*). The duration of the regulation is obtained through subtracting *WEF* from *UNT* and the activation time is obtained from subtracting *Published* from *WEF*. Finally the last two important characteristics of a regulation are the flight level affected and the reason for the regulation which will be referred to as the type of regulation.



Seq no	076	State	NEW
FMP	EHAAFMP	Published	26/05/2019 04:53
Regulations Id	EHAMA26	WEF	26/05/2019 07:00
Flight Level	ALL	UNT	26/05/2019 09:20
Reason	Aerodrome Capacity EHAM ARRIVALS ENTRY CROSS BORDER AIRSPACE		

Figure 2.4: Screenshot of ATFCM regulation message

During the post-operational phase, the impact of these regulations is assessed. Besides the characteristics available in the tactical phase, in the post-operational data for each regulation the number of planned and actually affected aircraft is given. In addition to that the planned and actual total ATFM delay caused by each regulation is also given. This data is stored in the NM Interactive Reporting (NMIR) database, however access to it is restricted to approved users only.

As it was mentioned the envisioned aim of the project is to be able to make predictions on the future state of the system, during the tactical phase of operations. To achieve this high-level objective the focus will be to predict the characteristics of future ATFCM regulations based on historical data and a period of observation during the day of operations.

[1] https://www.public.nm.eurocontrol.int/PUBPORTAL/gateway/spec/

3 Machine Learning for Predictions

Machine learning can be loosely defined as a collection of statistical methods that are aimed at learning from the input data. In classical programming, the programmer after having reviewed the data and determined the task he wants to achieve creates the set of rules with which the data will be processed. This set of rules and the data is then used as input and the output of the process is the answer to the task. Meanwhile, in the machine learning paradigm the data and the expected answers from the data are used as an input and the output are the rules with which the data has to be processed.

The large amount of data available in the aviation industry makes machine learning a very interesting approach for automating the process of analyzing this data. It has been proven that machine learning algorithms such as neural networks offer great capabilities in modelling non-linearities in the data. Based on these advantages, the machine learning approach is deemed appropriate for the purposes of this research project.

Extensive research in the available literature clearly indicated that there is a lack of studies aimed to predict the characteristics of European ATFCM regulations. The majority of literature considered in this study consist of finding a solution to the problem of predicting air travel delays. Most of the papers that will be discussed in this section focus on prediction of arrival and/or departure delay without much consideration on the source of the delay. As an exception [6], was aimed specifically to predict the delays caused by Ground Delay Programs (US equivalent of ATFCM regulations). Finally all the papers reviewed consisted of applying supervised learning methods in order to make their predictions.

The papers that will be discussed in the following sections are grouped based on the machine learning methods used. In section 3.1, papers that utilized ensemble learning are discussed and in section 3.2 papers that utilized neural network architectures are treated. For each paper that will be referenced, the discussion will start with the purpose of the paper followed by the algorithms and the inputs used in the study and finally the discussion of each paper is concluded by showing the respective results. In Table 3.1 an overview of the papers considered in this chapter is given.

3.1. Ensemble Methods

A significant amount of literature studied, as can be seen in Table 3.1, utilized ensemble methods in order to solve classification or regression problems. Ensemble learning consists of building a learning model on top or in parallel with other learning models[14]. Such methods typically make use of homogeneous learning models such as decision trees as was done in [6], [41] and [26]. It is also possible to utilize heterogeneous learning methods where the output of one learning model is used as input for another model as it was done in [47] and [21].

3.1.1. Ensembles of heterogeneous learners

Paper 1 - Introduction

In [21] Kim et al. utilize two deep learning models for the task of predicting the delay of flights operating to and from ten of the major US airports, without making a distinction between arrival or departure delay. In the first stage of prediction the authors use a deep recurrent neural network (RNN), to predict the class of daily delay status at one of these airports. The authors hypothesize that the accuracy of a RNN increases with depth. This daily delay status is defined to be a binary variable indicating if delays are occurring or not at the airport, based on a threshold value. The thresholds used in [21] were 15 and 30 minutes. In order to create a deep RNN architecture the authors present four different alternatives, which are listed bellow together with the authors argumentation for each.

- **Deep input-to-hidden** - Offers the effect of non-linear dimensionality reduction, allowing for extraction of the most important features.
- **Deep hidden-to-output** - Can offer advantage of disentangling the factors of variation in the hidden states, allowing for an easier prediction
- **Deep hidden-to-hidden** - Allows the RNN to learn highly non-linear transitions between consecutive hidden states
- **Stack of hidden states** - Allows the model to capture state transitions of different time scales.

Paper 1 - Methodology

The final chosen architecture consisted of deep input-to-hidden layer, followed by a stack of Long Short-Term Memory (LSTM) cells and a hidden-to-output layer, for a visual depiction of the model please refer to [21]. To train the model on time performance data of commercial airline flights and weather data were obtained. The data was sorted by airport so that they can be used as input for the first stage. For testing the algorithms the authors focused on the airport of Atlanta and flights to and from it. All departure and all arrival delays for a single day were averaged respectively. This average value is the representation of the delay status of a single day. The weather data was also averaged for a day. Based on the above input the class of delay is computed and this classification is repeated for consecutive days.

Once the daily delay status is computed it is used as input to a layered neural network that aims to predict for individual flights if they are delayed or not, based on the same thresholds of 15 and 30 minutes. Other inputs for this second stage include time information for the flight, delay statuses of the origin and destination airports and weather data. The authors do not mention how far into the future they predict the delay of the flights. However since the daily delay status is predicted for the day of operations it can be assumed that this prediction is performed for all flights scheduled to operate to and from Atlanta on the day of operations.

Paper 1 - Results

The results of the different experiments conducted with the model created in [21] are given in Figure 3.1. For the deep RNN model two length of sequences were tested. In the first case the input sequence was 7 days with the delay threshold set at 15 minutes and in the second case the sequence was 9 days with the delay threshold set two 30 minutes. No reasoning has been given for the selection of these particular combinations or why the authors did not test all 4 possible combinations. The proposed deep RNN, that has been called "Combined" in Figure 3.1, is compared to its constituent elements and a shallow model consisting of an LSTM that is connected to an input and output layer. It can be seen from the left table of Figure 3.1 that deep RNN architecture indeed improves the accuracy, with the highest accuracy corresponding to the combined model. Comparison of the two sets of sequence length and threshold indicates that the model performs better on a longer sequence and higher threshold. With regards to the value of the threshold it is understandable that the results improve due to larger noise when considering a smaller threshold. However, it is unclear whether the improved accuracy is due to the longer sequence, the larger threshold or a combination of both.

The results for the experiments conducted in the second stage of the model, namely the layered neural network, are shown in the right side of Figure 3.1. For this case the topology of the network was varied in terms of number of layers, number of nodes in each layer, batch sizes and number of epochs. The highest accuracy was 87.42% with a network containing five hidden layers and trained with the mini-batch gradient descent algorithm.

Paper 2 - Introduction

In a somewhat divergent manner from the trend of topics considered for this study, in [47] authors of the paper consider the problem of predicting conflict between a pair of flying aircraft. The problem is a typical classification problem, with the output being a binary variable indicating conflict or not. The relevance of this paper with respect to this research project lies in the fact that for the problem of conflict detection it is very important to have minimal if not any false-negative errors. Similarly for predicting the future occurrence of ATFCM regulations it is deemed equally important to have minimal false-negatives. In order to solve the problem the authors, in a similar manner to the paper discussed above, consider a two stage approach. The proposed model is aimed at increasing classification accuracy and reducing false alarm rate.

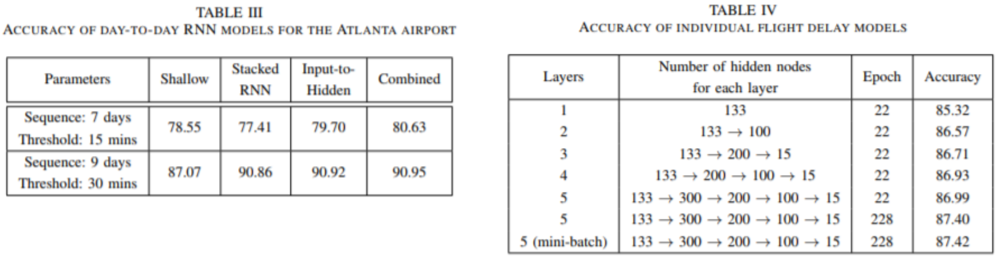


TABLE III
ACCURACY OF DAY-TO-DAY RNN MODELS FOR THE ATLANTA AIRPORT

Parameters	Shallow	Stacked RNN	Input-to-Hidden	Combined
Sequence: 7 days Threshold: 15 mins	78.55	77.41	79.70	80.63
Sequence: 9 days Threshold: 30 mins	87.07	90.86	90.92	90.95

TABLE IV
ACCURACY OF INDIVIDUAL FLIGHT DELAY MODELS

Layers	Number of hidden nodes for each layer	Epoch	Accuracy
1	133	22	85.32
2	133 → 100	22	86.57
3	133 → 200 → 15	22	86.71
4	133 → 200 → 100 → 15	22	86.93
5	133 → 300 → 200 → 100 → 15	22	86.99
5	133 → 300 → 200 → 100 → 15	228	87.40
5 (mini-batch)	133 → 300 → 200 → 100 → 15	228	87.42

Figure 3.1: Results of the experiments conducted in [21]

Paper 2 - Methodology

In the first stage of the proposed approach the authors consider using four different base classifiers in order to create a meta-dataset that will be used as input for the second stage. The classifiers considered for this stage are K-nearest neighbors (KNN), Naive Bayes Classifier (NBC), Back-Propagation Neural Network (BPNN) and Support Vector Machine (SVM). The outputs of the four base classifiers are respectively: the number of samples belonging to positive and negative classes, the conditional probability of a conflict, the weight of each prediction and distance from points to hyperplane.

Once the first stage produces outputs, these are used to train a second stage classifier. The second stage chosen by the authors is a SVM arguing that SVM offers generalization advantages and avoids getting stuck on local minima. The SVM utilized in the second stage classification has been modified to output a probability of conflict instead of an output $\in \{1, -1\}$. If this output probability is higher than a threshold then there is a conflict. The threshold in this paper was chosen to be 50%.

In order to train the model a dataset containing positions and velocities of aircraft, the lookahead times and in case of turning flights the turning moments and turning angles is used. The dataset is split into three parts. The first part is used for training the base classifiers. The second dataset is used for testing these base classifiers. From the test the output of the base classifiers is used to construct a meta-dataset that is used to train the second stage classifier. Finally the third part of the dataset is used to test the whole ensemble. Due to the fact that conflicts have a low rate of occurrence, the non-conflict samples predominate in the collected data. In order to deal with this class imbalance, the authors use the Synthetic Minority Over-Sampling Technique (SMOTE). Furthermore the authors normalize the magnitude differences between the features through the use of the Min-Max Normalization technique.

Paper 2 - Results

The results of testing each base classifier individually are shown in Figure 3.2. From the results presented in Figure 3.2 it can be seen that the base classifiers can reach an accuracy between 80 to 90%. The false alarm rate (conflict predicted when there was no conflict) ranges between 12 to 20%. Finally the missing alarm rate (no conflict predicted when there was conflict) is about 10% with the exception of SVM that was able to find all conflicts. This result makes the SVM particularly suitable for using it in the second stage classification. When testing the full ensemble on 100 straight segment flights the classification accuracy reported by the authors is 97% and the false alarm rate is 4.05%. While testing turning segment flights the classification accuracy dropped to 91% with the false alarm rate at 22.5% and no missing alarm.

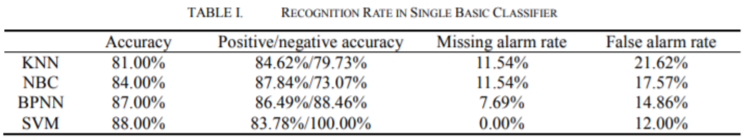


TABLE I. RECOGNITION RATE IN SINGLE BASIC CLASSIFIER

	Accuracy	Positive/negative accuracy	Missing alarm rate	False alarm rate
KNN	81.00%	84.62%/79.73%	11.54%	21.62%
NBC	84.00%	87.84%/73.07%	11.54%	17.57%
BPNN	87.00%	86.49%/88.46%	7.69%	14.86%
SVM	88.00%	83.78%/100.00%	0.00%	12.00%

Figure 3.2: Results of the base classifiers tested individually in [47]

3.1.2. Ensembles of homogeneous learners

Paper 1 - Introduction

In [41] the authors consider the problem of predicting departure and arrival delay of commercial flights in the US. In a similar manner to the papers discussed in subsection 3.1.1, the model proposed in this paper consists of a staged approach. In the first stage a classification is performed to predict if a flight is going to be delayed or not. The authors have used the convention utilized by the Bureau of Transportation Statistic and define a flight to be delayed only if the delay exceeds 15 minutes. If the classification stage predicts that a flight will be delayed then in the second stage a regression is performed to estimate the value of the delay. The machine learning models considered for each of the stages are listed bellow.

- Classification:
  - Gradient Boosting Classifier
  - Random Forest Classifier
  - Extra-Trees Classifier
  - AdaBoost Classifier
- Regression:
  - Extra-Trees Regressor
  - Random Forest Regressor
  - Gradient Boosting Regressor
  - Multilayer Perceptron

Paper 1 - Methodology

In order to train the model a dataset containing 12 features related to the on time performance(OTP) of domestic United States flights over the course of 5 years (2012-2016) was collected. Weather data was also collected for the same periods under consideration. The geographic scope of the data was limited to 15 of the major US airports. In order to remove irrelevant features the authors performed feature selection using the *Recursive Feature Elimination* algorithm. The other procedures for feature selection investigated by the authors include, *Univariate Tree-based* feature selection and *Principal Component Analysis*. The result of the feature selection process was that for prediction of departure delays 9 airline OTP and 12 weather features were selected. For the arrival delay 12 airline OTP and 24 weather features were selected. The reason for the higher number of features in arrival delay prediction is reported to be the fact that departure features have a significant impact on arrival delay. In terms of prediction accuracy feature selection caused only a 0.2% increase from 91.63% to 91.86%, but in terms of training time there was 56% decrease in time needed from 66 seconds to 37 seconds.

Paper 1 - Results

During training for the classification stage, the authors observed that the dataset was imbalanced. In order to correct for this imbalance initially the authors reduced the number of airports considered from 15 to 10 so that the ratio of minority to majority was maximised. However it was found that this was not enough to balance the dataset, as a result of this sampling of the data had to be performed. The sampling techniques investigated include *Random Undersampling*, *SMOTE* and *SMOTE + Tomek links*. The best performance was reported to be for *SMOTE + Tomek links* where the classification accuracy increased from 91,86% to 94% in the case of the Random Forest Classifier.

For evaluation of performance of the regression stage the mean square error (MSE) was selected as the metric. To reduce the MSE firstly the features were scaled to account for the variation of scales in the dataset. Two techniques were investigate for this: *Robust Scaler* and *Standard Scaler*. In the case of the Random Forest the original MSE was 102.18, with application of the *Standard Scaler* the MSE reduced to 90.34 and with *Robust Scaler* the MSE was 76.31. Another step to reduce the errors of the model was hyper-parameter tuning, which was done through the use of a *Grid Search*. Finally *Selective Training* was performed, where instead of training the model on the entirety of the origins and destinations the model was trained on individual OD pairs. The results of the paper are given in Figure 3.3. It can be seen from Figure 3.3 that for classification Gradient Boosting performs the best in both arrival and departure delay classifications. Meanwhile for the regression stage Extra-Trees offers the best performance. For both classification and regression stages the results for arrival delay prediction are better than those for departure delay, no explicit explanation has been stated from the authors for this.

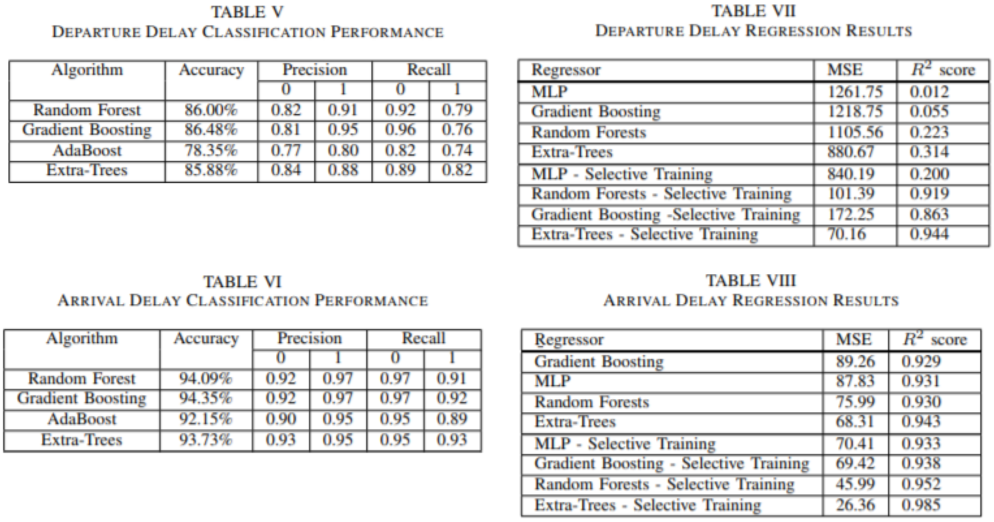


TABLE V
Departure Delay Classification Performance

Algorithm	Accuracy	Precision		Recall	
		0	1	0	1
Random Forest	86.00%	0.82	0.91	0.92	0.79
Gradient Boosting	86.48%	0.81	0.95	0.96	0.76
AdaBoost	78.35%	0.77	0.80	0.82	0.74
Extra-Trees	85.88%	0.84	0.88	0.89	0.82

TABLE VI
Arrival Delay Classification Performance

Algorithm	Accuracy	Precision		Recall	
		0	1	0	1
Random Forest	94.09%	0.92	0.97	0.97	0.91
Gradient Boosting	94.35%	0.92	0.97	0.97	0.92
AdaBoost	92.15%	0.90	0.95	0.95	0.89
Extra-Trees	93.73%	0.93	0.95	0.95	0.93

TABLE VII
Departure Delay Regression Results

Regressor	MSE	R^2 score
MLP	1261.75	0.012
Gradient Boosting	1218.75	0.055
Random Forests	1105.56	0.223
Extra-Trees	880.67	0.314
MLP - Selective Training	840.19	0.200
Random Forests - Selective Training	101.39	0.919
Gradient Boosting -Selective Training	172.25	0.863
Extra-Trees - Selective Training	70.16	0.944

TABLE VIII
Arrival Delay Regression Results

Regressor	MSE	R^2 score
Gradient Boosting	89.26	0.929
MLP	87.83	0.931
Random Forests	75.99	0.930
Extra-Trees	68.31	0.943
MLP - Selective Training	70.41	0.933
Gradient Boosting - Selective Training	69.42	0.938
Random Forests - Selective Training	45.99	0.952
Extra-Trees - Selective Training	26.36	0.985

Figure 3.3: Results of models tested in [41]

Paper 2 - Introduction

In [26] the authors consider the same problem as in the paper discussed above ([41]), so prediction of arrival and departure delays. In contrast to [41], in [26] the problem has been posed only as a single regression stage problem utilizing Gradient Boosted Decision Trees. Gradient Boosting is a procedure where starting from an initial learning unit, a successor unit is employed to improve upon the errors of the predecessor.

Paper 2 - Methodology

The data used in [26] consist of flight delay data over the periods between April 2013 to October 2013. The dataset covers flights taking place to or from 70 of the busiest airports in the US. Initially 14 features were present in the dataset, however after applying some exploratory statistics only 8 features were selected as input for the model. The selection criteria for the features to be used is stated by the authors to be the features that showed the highest correlation factor. The features used as input consist of: day of week, carrier, origin/destination airport, scheduled departure/arrival time and arrival/departure delay. The last two features are used as the supervisory signal. In terms of pre-processing the features were normalized on a uniform scale of 0 to 1 and a mean of 0. In addition, the data was inspected for outliers and such outliers were discarded. The way this was done was was to consider only points whose delay was in the range between $Q1-1,5IQR$ and $Q3+1,5IQR$, with $Q3$ and $Q1$ being the 75 and 25 percentile respectively and IQR the inter-quartile range.

Paper 2 - Results

The Gradient Boosted model constructed consisted of the hyper-parameters that are listed bellow. For evaluation of the model three metrics were selected: Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Coefficient of Determination (R^2). The results of this paper are presented in Figure 3.4. Comparing the results of Gradinet Boosted model in [41] and [26], the results seem to be in the same order of magnitude for the arrival delay. The results are considerably different for departure delay with the model of [26] performing better.

- Max number of leaves per tree was 8
- Min number of samples per leaf node was 10
- Learning rate was 0,05
- Total number of trees was 1000

Paper 3 - Introduction

Finally the last paper considered for this study in the category of models utilizing homogeneous ensembles consists of [6]. The purpose of this paper is to predict the performance of ground delay programs (GDP).

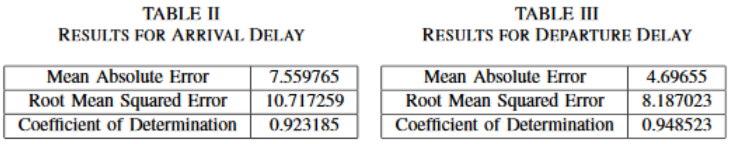


TABLE II
RESULTS FOR ARRIVAL DELAY

Mean Absolute Error	7.559765
Root Mean Squared Error	10.717259
Coefficient of Determination	0.923185

TABLE III
RESULTS FOR DEPARTURE DELAY

Mean Absolute Error	4.69655
Root Mean Squared Error	8.187023
Coefficient of Determination	0.948523

Figure 3.4: Results for arrival and departure delay prediction in [26]

GDPs are the US equivalent to the European ATFCM regulations. In essence once a demand capacity imbalance is identified, the A/C relevant to the imbalance are issued a delay on ground by the NMOC in Europe or Air Traffic Control System Command Center (ATCSCC) in the US [38]. As such [6] has a lot of relevance for the purposes of this literature study.

The authors of [6] are interested to predict the performance of a planned GDP. In order to quantify this performance they want to predict the average arrival delay and number of cancelled arrivals that will be caused by this GDP. They aim to achieve this through identifying historical GDPs applied on conditions similar to the day that the prediction is to be made on. The proposed model in the paper is a variation of Geographically Weighted Regression (GWR). In GWR a prediction is wanted for a particular location l_p. To make this prediction observations that occurred in surrounding locations l_i, where l_i is the i^{th} location around l_p, are taken into account. The basic assumption is that observations that occurred closer to l_p are more relevant than others further away. To achieve this weights are assigned to the observations based on a distance metric and chosen kernel function. For the purposes of this study the Gaussian kernel as shown bellow was used to generate the weights.

$$w_i(d,\beta) = \exp(\frac{-d(l_p,l_i)^2}{\beta}) \tag{3.1}$$

In Equation 3.1, $d(l_p,l_i)$ is a function evaluating the distance between location of prediction l_p and location of observation l_i, β is a parameter called the bandwidth. As β goes to infinity the weights of observations will tend to 1. In typical GWR the distance function usually implies geographic distance, however for the purposes of [6] this distance metric has been adapted to be related to similarity in traffic and weather conditions between different days.

Paper 3 - Methodology

The authors have assumed that all GDPs in the considered dataset to have occurred in a single airport. They utilize the demand and terminal weather to estimate the capacity distribution on the historical days. Comparison of the estimated capacity distribution on different days and the day of prediction is used as a measure of distance between the observations and using Equation 3.1 the weights are assigned. The authors note that during the day of operations the demand and weather conditions are only forecasted and are not known with certainty as is the case for the previous historical days.

Target variables & loss functions As it was mentioned, the performance of GDPs is quantified in this paper through the arrival delay and number of cancelled arrivals. The authors are interested to find the average values for the above quantities and also the 90^{th} quantiles as a measure of the worst case scenario. As a result of these two objectives, two different loss functions are considered. For the problem of predicting the average values of arrival delay and number of cancelled arrivals, the chosen loss function was the absolute error. It is argued by the authors that the absolute error leads to models that are more robust to outliers. For the problem of predicting the 90^{th} quantiles the chosen loss function is given as follows.

$$f(y,\hat{y}) = \begin{cases} (\alpha-1)(y-\hat{y}) & \text{if } y < \hat{y} \\ \alpha(y-\hat{y}) & \text{if } y > \hat{y} \end{cases} \tag{3.2}$$

In Equation 3.2, y is the observation, $\hat{y}$ is the prediction. Equation 3.2 is said to be minimized when $\hat{y}$ is equal to the α-quantile of y. Thus using this function an estimate of the quantile is obtained. The chosen machine learning models for this study were Random Forest and Gradient Boosted Forest. In both methods the prediction comes from a collective of decision trees.

Random forests In Random Forest each tree is constructed independently of the others and randomness introduced in the learning process with the aim of reducing the variance of predictions. The observation weights calculated with Equation 3.1 are taken into account in the splitting criteria of the leaf nodes and in the end prediction once the forest is created. A disadvantage of random forests is that the method does not easily permit usage of arbitrary loss functions, as such random forest is not considered for the 90^{th} quantile prediction problem.

Gradient Boosting In Gradient Boosted Forests the trees are built sequentially with each new tree trying to compensate for the errors of the previous. They allow the usage of any loss function, as such it will be used for both average value and 90^{th} quantile prediction problems. The authors state that weighing of observations can be done by including the weights in the splitting criteria of each tree or directly in the loss function. However it is not clear how the authors implemented the weights.

Comparison models For comparison purposes the two forest methods utilizing the proposed weighing strategy will be compared to 4 baseline models. The first baseline was taking a weighted average of all the observation, with the distance measure as described above. The second baseline was using KNN and using the average of the k observations. Finally the Random Forest and Gradient Boosted Forest were considered without the weighing scheme proposed. The last two were named the global methods and the proposed ones with the weighing scheme were named the spatial methods.

The features of the data or the explanatory variables considered for making the prediction consist of the following:

- Entry time - time GDP was declared and put in effect, measured in minutes after 4:00 am
- Earliest ETA - the earliest arrival time for flights affected by GDP, measured in min after 4:00 am
- Duration - Difference between earliest and latest arrival of flights affected by GDP, in minutes
- Airport Acceptance Rate(AAR) in time period t - A period t is defined as a 15 minute interval starting from 4:00 am. If during the period t a GDP was planned the AAR was recorded, otherwise it was left undefined.
- Average AAR - Average AAR in the time interval that GDP was planned
- Number of Core 30 airports - Number of main 30 airports that were affected by GDP
- Ground Stop duration - Duration of the ground Stop that led to GDP. If none then it was set to 0

Validation In order to obtain the most accurate results possible the parameter tuning was performed. For the Random Forest the only parameter that was tuned was the number of trees in the forest the rest of the parameters were left to the default scikit-learn parameters. For the Gradient Boosted Forest the number of trees, maximum depth and learning rate was tuned. Furthermore the parameters for the bandwidth β and k in KNN had to be tuned. To select these parameters the authors used the leave-one-out validation technique. The approach consists to fit the model several times to the dataset with each time removing one of the observations from the data. The resulting model from each fit is used to make a prediction and the loss is averaged. The parameters that resulted in the lowest average loss were selected.

Parameter tuning For the Random Forest model the number of trees was first tuned for the global variant. The authors assumed that the value for this parameter would work well also with the spatial model. The number of trees were steadily increased and the average leave-one-out loss was plotted against this number. The number of trees to be used was chosen as the point in the resulting plot where the curve became flat. For the Gradient Boosted Forest a grid search was performed for the global variant of the model. The learning rate was allowed to take values $\in \{0.2, 0.1, 0.05, 0.01, 0.005\}$, the maximum tree depth was allowed to take each value between 2-7, the number of trees was varied between 1-300. After the optimal hyper-parameters were found, the bandwidth in Equation 3.1 was tuned for the spatial variants of the models. The average leave-one-out loss was recorded for bandwidth values between 0,3 and 10 with a step of 0,05. For the KNN baseline a similar procedure was performed by recording the average leave-one-out loss while k was varied between 1 to 100.

Paper 3 - Results

The results of the proposed models for prediction of the average value and 90^{th} quantile of the arrival delay and number of cancelled arrivals is given in Figure 3.5. As it can be seen from the two top tables of Figure 3.5 the spatial variants that weigh the observation based on traffic and weather conditions outperform their global counterparts. For the problem of predicting the average arrival delay the spatial random forest performed the best, while for the prediction of average number of cancelled arrivals the spatial gradient boosted forest model performed the best. In the two bottom tables of Figure 3.5 it can be seen that the gradient boosted models outperformed the baseline methods, but there is only marginal improvement between the global and spatial variants.

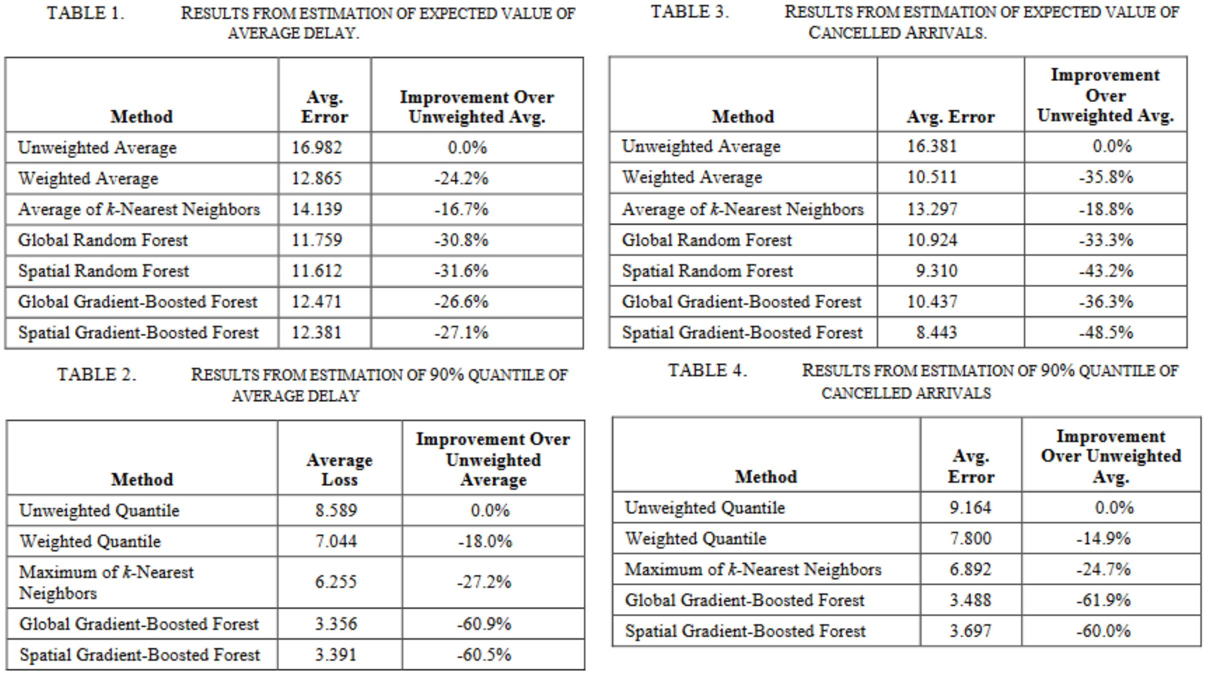


TABLE 1. RESULTS FROM ESTIMATION OF EXPECTED VALUE OF AVERAGE DELAY.

Method	Avg. Error	Improvement Over Unweighted Avg.
Unweighted Average	16.982	0.0%
Weighted Average	12.865	-24.2%
Average of *k*-Nearest Neighbors	14.139	-16.7%
Global Random Forest	11.759	-30.8%
Spatial Random Forest	11.612	-31.6%
Global Gradient-Boosted Forest	12.471	-26.6%
Spatial Gradient-Boosted Forest	12.381	-27.1%

TABLE 3. RESULTS FROM ESTIMATION OF EXPECTED VALUE OF CANCELLED ARRIVALS.

Method	Avg. Error	Improvement Over Unweighted Avg.
Unweighted Average	16.381	0.0%
Weighted Average	10.511	-35.8%
Average of *k*-Nearest Neighbors	13.297	-18.8%
Global Random Forest	10.924	-33.3%
Spatial Random Forest	9.310	-43.2%
Global Gradient-Boosted Forest	10.437	-36.3%
Spatial Gradient-Boosted Forest	8.443	-48.5%

TABLE 2. RESULTS FROM ESTIMATION OF 90% QUANTILE OF AVERAGE DELAY

Method	Average Loss	Improvement Over Unweighted Average
Unweighted Quantile	8.589	0.0%
Weighted Quantile	7.044	-18.0%
Maximum of *k*-Nearest Neighbors	6.255	-27.2%
Global Gradient-Boosted Forest	3.356	-60.9%
Spatial Gradient-Boosted Forest	3.391	-60.5%

TABLE 4. RESULTS FROM ESTIMATION OF 90% QUANTILE OF CANCELLED ARRIVALS

Method	Avg. Error	Improvement Over Unweighted Avg.
Unweighted Quantile	9.164	0.0%
Weighted Quantile	7.800	-14.9%
Maximum of *k*-Nearest Neighbors	6.892	-24.7%
Global Gradient-Boosted Forest	3.488	-61.9%
Spatial Gradient-Boosted Forest	3.697	-60.0%

Figure 3.5: Results of the proposed weighted methods and baseline methods in [6]

3.2. Neural network methods

Paper 1 - Introduction

In [20] Khanmohammadi et al. propose a new method for handling categorical input variables for a neural network. Neural networks are designed to work with numbers and tensors of numbers, as such these type of variables have to be transformed to be used in a neural network. The most used transformation is encoding. Encoding can be done through transforming a categorical variable into a single integer. This type of encoding can be problematic since it implies that a higher number would be connected to a higher importance which most of the times is not wanted. This problem is solved through one-hot encoding or as it is referred to in [20] "1-of-N" encoding is used. In one-hot encoding a categorical variable is replaced with a vector mostly filled with zeros and a one at a particular index. However it is noted by the authors that "1-of-N" encoding can introduce multicollinearity that can lead to an ill conditioned problem". An additional issue with one-hot encoding is that the input grows with the number of categories in the nominal variables adding to the complexity of the model.

Paper 1 - Methodology

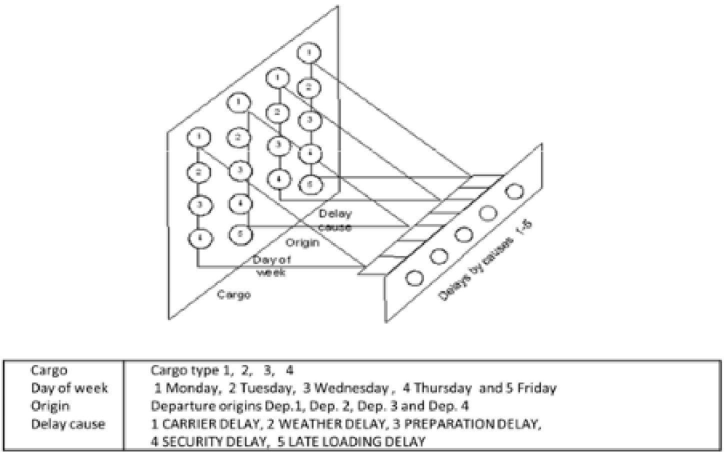
The authors propose a multilevel input layer to handle this issue. In a BPNN the input layer is composed of as many nodes as there are input features. In the proposed method in addition to each node for each input feature there are also as many nodes as there are categories of each input feature. This concept is illustrated in Figure 3.6. As it can be seen in Figure 3.6 one of the input features is cargo type and it has 4 different categories. Each of the nodes of the multilevel input layer is connected to all the nodes in the output layer.

The nodes in this input layer are binary neurons, they take value of 1 if they are active and 0 otherwise. The activation of the input nodes is indicative of that particular input being contributing at the end output.

Paper 1 - Results

The proposed architecture was used by the authors to predict the arrival delay of inbound flights at JFK. A dataset containing all the inbound flights to JFK for January 2012 was collected. The dataset contained around 1100 flights and the following features were used for training. The code of the airport was converted to an integer between 1 to 53 (total number of origin airports). The values for the delay reasons were normalized by dividing with the maximum value for each of the reasons and the model was trained for 10000 epochs. After training the model was tested on prediction of arrival delays for 5 flights. The RMSE of the proposed model for the normalized delay is reported to be 0.1366, compared to 0.1603 for a BPNN. The run time of the proposed method was 38% faster than a BPNN, while using 21% more memory than the BPNN. Due to the fact that the RMSE is reported unit-less it must be the RMSE for prediction of the normalized delay. For this reason the results of this paper can not be compared with the results of the previous studies.

- Inputs used in [20]
  - Day of month: 1-31
  - Day of week: 1-7
  - Code of origin airport
  - Scheduled/Actual departure time
  - Scheduled/Actual arrival time at JFK
  - Arrival delay
  - Reason 1: Carrier delay value
  - Reason 2: Weather delay value
  - Reason 3: National Air Space delay value
  - Reason 4: Security delay value
  - Reason 5: Late aircraft delay value



Cargo	Cargo type 1, 2, 3, 4
Day of week	1 Monday, 2 Tuesday, 3 Wednesday , 4 Thursday and 5 Friday
Origin	Departure origins Dep.1, Dep. 2, Dep. 3 and Dep. 4
Delay cause	1 CARRIER DELAY, 2 WEATHER DELAY, 3 PREPARATION DELAY, 4 SECURITY DELAY, 5 LATE LOADING DELAY

Figure 3.6: Illustration of a multilevel input layer proposed in [20]

Paper 2 - Introduction

In [31] Pamplona et al. utilize layered neural network to predict whether flights will be delayed or not. The flights considered in this study consisted of the flights between Congahonas airport in Sao Paulo and Santos Dumont airport in Rio de Janeiro. This route is chosen because it is the air route with the highest frequency of flights in Brazil. The reference period for creation of the dataset was January of 2017, during which 1560 flights took place between the two airports. A flight within this study is considered to be delayed if it arrives later than 15 minutes from the scheduled arrival time. The inputs used for this model are listed bellow.

- Inputs in [31]:

- Airline ID
- Day of the week
- Real departure block - Actual departure time. A block is a one hour interval, there are 17 blocks starting from 06:00 to 22:00
- Real arrival block - Similar to the above
- Departure delay - Delays are divided into 4 blocks. Block 1 contains delays between 0 to 15 minutes, block 2 delays between 15 to 30 minutes, block 3 delays between 30 to 60 minutes and block 4 delays above 60 minutes

Paper 2 - Methodology

In terms of neural network architecture the title of the paper suggests that the multilevel input layer presented in [20] is used in this study as well. However, within the paper utilization of such an input layer is never discussed. In order for hyper-parameterization the ranges for the parameters were chosen as follows: 1) epochs ∈ {5,10,100,1500}; 2) batches ∈ 5,10; 3) optimizers RMSProp, Adam; 4) activation functions ReLu, Sigmoid; 5) number of neurons in hidden layers ∈{1,2,3,4,5,6,7,8,9,10,20,30}; 6) number of hidden layers ∈{0,1,2,3,4,5}.

For selection of the optimal parameters the authors used the Random Search technique. This method is an alternative to the Grid Search where random sampling in the hyper-parameter space is done. This technique has been proposed by Bergstra and Bengio in [2], where they show that the Random Search technique can produce as good models as Grid Search at lower computational times. The parameters chosen with this technique were in validated through K-Fold validation.

Paper 2 - Results

The result of the hyper-parametrization process was Adams optimizer and 4 hidden layers. The number of neurons in each layer from the first to fourth were 10,4,10,20. The activation functions in the four hidden layers were ReLu, ReLu, Sigmoid, ReLu and in the output layer Sigmoid. The number of epochs was 10 and the batch size also 10. The accuracy of the resulting model was 91% and the resulting confusion matrix is presented in Figure 3.7.

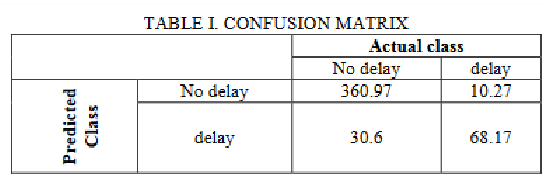


TABLE I. CONFUSION MATRIX

		Actual class	
		No delay	delay
Predicted Class	No delay	360.97	10.27
	delay	30.6	68.17

Figure 3.7: Confusion matrix of the model in [31]

Paper 3 - Introduction

In [11] Gopalakrishnan and Balakrishnan perform a comparative analysis of different machine learning methods for the task of predicting air traffic delays. They pose three problems on which the different methods will be compared on. In addition to the performance of the models on the three problems the authors have performed an investigation on the effect of different input feature vectors on the performance of the models. The three problems considered are listed bellow.

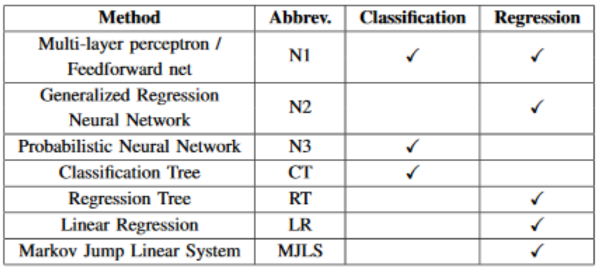
1. Classification of OD pair delays - Will the delay on an OD pair in the next Δt hours exceed a delay threshold?
2. Prediction of OD pair delay - What will be the value for the OD pair delay, Δt hours from now?
3. Prediction of airport delay - Similar to the OD pair delay

Paper 3 - Methodology

Out of the three problems, only the first two will be discussed in this literature study. The reason for this choice is that the second and third problems differ only slightly in terms of input vectors and the results in terms of best performing model for both problems are similar in ranking. The authors have defined an OD pair delay to be "for each hour of the day, the median departure delay of all flights that took off from that origin airport to the same destination airport during that hour". Through the OD pair delays the authors construct "delay networks", which are graphs whose nodes are the airports in the US and the edges represent the OD pair delays. They further use this representation to describe a day of operations as a time series of these delay networks, one for each hour of the day. In terms of the prediction horizon Δt they consider 2,4,6 and 24 hours and for the delay threshold they consider 30, 60 and 90 minutes. The potential inputs for the models considered are listed bellow. They are referred to as potential inputs because the authors select subsets of these inputs to create different input vectors, for the particular feature vectors that the authors consider please refer to [11]

- Temporal variables:
  - Time of day
  - Day of week
  - Season - Year is split into seasons based on delay values
- Local delay variables:
  - OD-pair delays - Current OD pair delay and also the delay for the past hours
  - Delay on adjecent OD pair
- Network delay variables:
  - Type of hour (delay mode) - This a variable that has been identified by the authors in a previous study through clustering of delay networks. An example of this type of variable is "at the current hour delays are increasing at ATL"
  - Type of day - Variable determined by grouping the sequence of delay networks into one of the six potential categories.

The models that are considered for the three problems posed in this study are given in Figure 3.8. All the proposed models for delay prediction are of the supervised learning type with the exception of the Markov Jump Linear System (MJLS). This last model is based on Markovian transitions of the current delay mode of the delay network. Through knowing the current delay mode and the system transition probabilities the future delay mode distribution is determined. For specifics of this model please refer to [11] and [12]



Method	Abbrev.	Classification	Regression
Multi-layer perceptron / Feedforward net	N1	✓	✓
Generalized Regression Neural Network	N2		✓
Probabilistic Neural Network	N3	✓	
Classification Tree	CT	✓	
Regression Tree	RT		✓
Linear Regression	LR		✓
Markov Jump Linear System	MJLS		✓

Figure 3.8: Models considered in [11]

Paper 3 - Results

For the classification problem with prediction horizon of 2 hours and a threshold of 60 min the highest accuracy is reached for the N1 model with input vector F2 (current OD pair delays and time of the day). It was found that the neural network methods outperform the classification tree and for the feed-forward network the input features do not lead to significant changes in accuracy. Choosing the N1 model with the F2 vector, the authors investigated the effect of different delay thresholds on the prediction accuracy. The average accuracy for 30 minute threshold is 85% and as the threshold increases the accuracy increases to 97% for a threshold of 90 minutes.

For the regression problem out of the supervised learning methods considered it was found that the best performing models are the N2 and RT with input vector F7 (OD pair delays at the current hour, delay on adjacent OD pairs and time of day). Comparing the best performing supervised learning methods with the MJLS the authors found that MJLS outperforms N2 and RT. Both the mean OD pair delay error and standard deviation of the prediction errors is lower for MJLS compared to N2 and R7. The prediction error for the supervised models seems to be in accordance with the models that were seen in section 3.1.

Finally the effect of prediction horizon Δt on the prediction accuracy is investigated. For the classification problem the authors report that the prediction accuracies decrease by less than 1% when Δt is increased from 2 hours to 4,6 or 24 hours. These results seem to be in agreement with the results of [24], where the authors tried to predict if a GDP would be initiated by using logistic regression for prediction horizons of 1,2,3 and 4 hours. For the regression problem the authors found that the performance of the MJLS model remained constant when the prediction horizon was increased, while the performance of the generalized regression neural network and the regression tree severely degraded.

Ref.	Title	Year	Problem type	Problem question	Focus	Inputs	Model(s)	Accuracy	Error
[47]	Application of ensemble learning algorithm in aircraft probabilistic conflict detection in free flight	2018	Classification	Given features of two aircraft in straight/ turning segments, will the pair be in conflict?	Classification accuracy	-Positions -Velocities -Look ahead times -Turning moments -Turn angles	2 step ensemble 1st stage: KNN,NBC,BPNN,SVM 2nd stage:SVM	91% (turning flights) 97% (straight flights)	N.A
[31]	Supervised neural network with multilevel input layers for predicting air traffic delays	2018	Classification	Will a flight be delayed more than 15 minutes?	Classification accuracy	-Airline ID -Day of week -Actual departure block time -Actual arrival block time -Departure delay (4 cases)	Layered neural network	91%	N.A
[24]	Predicting initiation of ground delay programs	2017	Classification	Will there be a GDP in k hours?	Feature identification	-Actual/Scheduled demand -Actual/Scheduled capacity -Actual/Forecasted weather	Logistic Regression	38-46%	N.A
[41]	A machine learning approach for prediction of on time performance of flights	2017	Classification + Regression	Will a flight be delayed, if so what will be the value for the delay	Comparison	-On-time performance data -Weather data	Classification: -Gradient Boosting -Random Forest -Extra Trees -AdaBoost Regression: -Extra Trees -Random Forest -Gradient Boosting -Layered neural network	86.48%/94.35%	70.16/26,36 (MSE)
[26]	A statistical approach to predict flight delay using gradient boosted decision tree	2017	Regression	What will be the value of arrival and departure delay of flights?	Regression accuracy	-Day of week -Airline -Origin/Departure airport -Scheduled arrival/departure time	Gradient Boosted Trees	N.A	4.69/7.56 (MAE) 8.19/10.72 (RMSE)
[11]	A comparative analysis of models for predicting delays in air traffic networks	2017	Classification / Regression	-What models are best for classification? -What models are best for regression? -What is the influence of different feature vectors ?	Comparison	-OD delays -Time of day -Day of week -Season -Type of day -Type of hour	Classification: -Layered neural network -Probabilistic neural network -Classification Tree Regression: -Layered neural network -Generalized neural network -Regression Tree -Linear regression -Markov Jump Linear System	Refer to paper	Refer to paper
[6]	Predicting performance of ground delay programs	2017	Regression	What is the average arrival delay and average number of canceled arrivals?	Innovation	-Airport acceptance rate(AAR) -Average AAR during GDP -Nr. of airports affected by GDP -Ground Stop Duration that led to GDP	Random Forest	N.A	11.61* (MAE)
[10]	Multivariate aviation time series modelling: VARs vs LSTMs	2016	Regression	Are VARs or LSTMs better suited for performing one-step/multi-step ahead forecasting?	Comparison	-Aircraft sensor data -VAR generated time series -LSTM generated time series	-Vector Autoregressive -Standard LSTM -Encoder-Decoder LSTM	N.A	Refer to paper
[20]	A new multi-level input layer artificial neural network for predicting flight delays at JFK	2016	Regression	What is the value of arrival delay for a flight?	Innovation	-Day of month/week -ID code origin airport -Scheduled/actual departure time -Delay at departure -Scheduled/actual arrival time	Multi-level input layer artificial network	N.A	0.14 (RMSE)
[21]	A deep learning approach to flight delay prediction	2016	Classification	What is the delay status for the airport and for individual flights what is the class of delay?	Classification accuracy	1st stage: -Weekday, Season,Month,Date -Day average weather data 2nd stage: -Weekday,Season,Month,Date -Origin/Destination airport -Scheduled departure/arrival time -Delay status at origin/destination -Weather data	1st stage: Deep RNN (deep input-to-hidden, stacked LSTMs, deep hidden-to-output) 2nd stage: Layered neural network	90.95% (day status) 87.42% (individual flights)	N.A

Table 3.1: Overview of the papers utilizing machine learning to make predictions that were considered in this study. When the performance values have a '/' in between, the first value represents departure and the second arrival delay prediction. When these values have a '-' they represent a range. For the gray shaded cell performance is presented in terms of the conditional probability of predicting GDP initiation when it actually occured. The value with the '*' represents only performance for arrival delay.

4
Nonlinear system analysis

In previous chapter it was seen that a commonality of several studies was that they considered making their predictions on different prediction horizons. In [11] it was found that depending on the chosen supervised learning model chosen, the prediction performance for the regression problem degrades with increasing prediction horizon. For the classification problem the authors of [11] and [24] also found a reduction in accuracy, albeit smaller than that of the regression problem. In the following sections of this chapter the theoretical factors that influence the choice of the prediction horizon will be discussed. The chapter is going to be finalized with a discussion of machine learning papers that have utilized the non-linear features of time series that are to be discussed in order to increase prediction accuracy.

4.1. Return intervals

One of the most important parameters to be taken into account when creating stochastic models for modeling random events is the return interval of the events under consideration. Other synonyms used for return interval include return periods or inter-arrival times depending on the field of study. A return interval is defined to be the time between the start or occurrence of two random events. When the events to be modeled are assumed to occur independently of each other the return intervals r follow the exponential distribution [3] [36] as

$$f(r) = \frac{1}{\langle r \rangle} e^{-\frac{r}{\langle r \rangle}} \tag{4.1}$$

where $\langle r \rangle$ is the mean return interval. In the study of extreme phenomena and rare events one is in particular interested in determining the return intervals of events that exceed a certain threshold q. Consider as an example the return intervals of ATFCM regulations that cause a total delay of q or above. Under the assumption that events are uncorrelated with each other, the mean return interval of events exceeding a threshold q, R_q is determined as follows [3]

$$R_q = \left(\int_q^\infty f(x) \right)^{-1} \tag{4.2}$$

where $f(x)$ is the probability density distribution of the events. The integral in the above expression represents the probability that a random event X with density distribution $f(x)$ will exceed q, $P(X \geq q)$. This probability is a measure of the occurrence rate of the extreme event and the mean return interval of such an extreme event is the reciprocal of the occurrence rate. Using the exponential distribution for the inter-arrival times and the Poisson distribution for the counts of events is useful to model systems that appear to have the memory-less property. That is that a certain probability distribution is independent of its past history. This property is formulated as follows

$$P(R > t + s | R > t) = P(R > s) \tag{4.3}$$

where R is a random variable with density distribution $f(r)$. The fact that R has survived up until t is not in particularly useful to predict if R will survive for $t + s$, thus the system has forgotten of the past. However, most physical process do not show the memoryless property. Instead the events are correlated with eachother and often they are best described by heavy tailed distributions[19].

4.2. Effect of long-range persistence on return intervals

As it was mentioned most natural phenomena have been shown to be correlated and these correlations can extend to very large time spans. Examples of such phenomena include river flows [16], stock and foreign exchange markets [46] [34], temperature records [3] [42] and also air traffic flows [45].

In [3] Bunde et al. analysed the return intervals for time series that show long-range persistence. Long-range persistence of time series has been defined in [3] and [36] to be present when the autocorrelations of such series show a power-law decay as

$$ACF(k) \sim k^{-\gamma} \;, 0 < \gamma < 1 \tag{4.4}$$

where ACF is the autocorrelation function, k represents the time lag between observations and γ is is defined in [36] as the autocorellation exponent. In Figure 4.1 the time series of the volatility (magnitude of fluctuations) of IBM stock on the 10^{th} of May 2002 is given. In this figure examples of return intervals (r_1, r_2, r_3) for different thresholds are shown. For each threshold value q there are N_q return intervals, r_q^i where $i = 1, ..., N_q$. In [3] it is stated that for the case where $N_q \gg 1$ the following holds.

$$\sum_{i=1}^{N_q} r_q^i \cong N \tag{4.5}$$

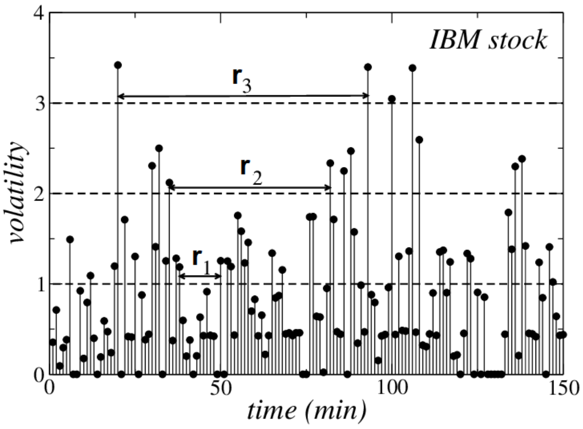


Figure 4.1: Illustration of return intervals for different thresholds adapted from [46]

In case the data is shuffled the long range dependencies are destroyed, however the above equation is still valid. As a result of this, for both correlated and uncorrelated data the mean return interval for a particular threshold R_q, can be evaluated as follows.

$$R_q \equiv \frac{1}{N_q} \sum_{i=1}^{N_q} r_q^i \cong \frac{N}{N_q} \tag{4.6}$$

Since the mean return interval for events above a certain threshold is unaffected by correlations in the data, it can be evaluated straight from the tail of the density distribution of the events as shown in Equation 4.2. The authors of [3] go on further to analyze the distribution of the return intervals in long term correlated data $P_q(r)$.

In order to determine the distribution of the return intervals $P_q(r)$ as a function of correlation exponent γ, the authors have generated records of length $N = 2^{21}$ for various values of γ by a technique involving the Fourier transform. For each γ, $P_q(r)$ was calculated for several thresholds q. In Figure 4.2 the distribution of the return intervals for the case where $\gamma = 0.4$ and $q = 2 (R_q \simeq 44)$ is given shown in the shaded gray plot. The straight line is the distribution of the return intervals when the data is shuffled, following the Poisson distribution. A considerable difference between the two distributions can be seen, in the correlated data the probability of having return intervals well bellow and well above R_q is higher. The authors conclude that the distribution of return intervals for long term correlated data behaves as follows

$$P_q(r) \sim e^{\left(-\frac{r}{R_q}\right)^{\gamma}} \tag{4.7}$$

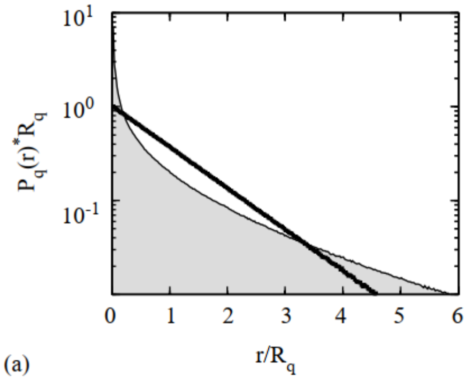


Figure 4.2: Distribution of return intervals for long term correlated data (gray shaded plot) and shuffled data (black line) obtained from [3]

In a final step in [3], the autocorrelation function of the return intervals $C_r(s)$ was evaluated. The results of this process are shown in Figure 4.3. In Figure 4.3 the results for $\gamma = 0.4$, $\gamma = 0.7$, $q = 1$ and $q = 2$ are shown. The curves for same values of γ are parallel straight lines, suggesting that the return intervals are also long-term correlated with the same exponent γ. The slope of the lines in Figure 4.3 reflects the value of γ. It can be seen from inspection of this figure that for a particular correlation exponent γ increasing the threshold q will lead to down-ward shift of the autocorrelation curve. This suggests that the behaviour of return intervals for thresholds that are seen rarely in real-life data can be inferred from the return intervals of thresholds present in the data.

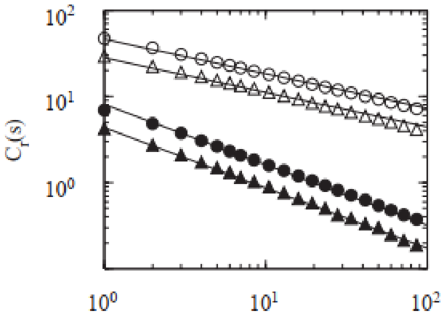


Figure 4.3: Auto-correlation function of the return intervals for $q = 1$ (circles) and q=2 (triangles) for $\gamma = 0.4$(open symbols) and $\gamma = 0.7$ (filled symbols). Obtained from [3]

In [44] Vera-Valdes argues that not taking into consideration the long range dependencies in data can lead to erroneous forecasts. Modelling long-range correlated data with standard methods will often lead to underestimations. These standard methods often rely on the applicability of the Central Limit Theorem (CLT). In [9] Fowler, lists the three conditions for the applicability of the CLT. Of particular importance is the condition that states that CLT is valid for random variables that have finite mean and variance. As it is mentioned in [19] and [35] a potential source of these long spanning correlations can be due to variables originating from heavy-tailed distributions with infinite variance. Fowler[9] gives as an example the case of the "Long-Term Capital Management" hedge fund and how their wrong assumption on the applicability of the CLT in 1998 almost brought a collapse in the global financial markets [1]

[1] https://www.investopedia.com/terms/l/longtermcapital.asp -Accessed on May 2019

4.3. Estimation of persistence

The most common technique used to quantify long-range persistence in a time series consists of the estimation of the Hurst exponent. The Hurst exponent H arises in self similar processes and it is a measure of the global level of persistence of the series [28]. Its value varies between 0 and 1. If $H = 0.5$ then the time series is similar to a random walk, if $0 < H < 0.5$ the series is anti-persistent (negatively correlated) and if $0.5 < H < 1$ the series is persistent (positively correlated)[48]. As H approaches 0 or 1, the anti-persistence and persistence respectively get stronger.

A self similar process is a stochastic process in which the behaviour of the process remains the same irrespective of scaling in time or space [19]. Consider a stochastic process $X(t)$, this process is said to be self similar if the following condition holds

$$X(at) \overset{d}{=} a^H X(t) \tag{4.8}$$

where a is a scaling factor such that $a > 0$, H is the Hurst exponent and the symbol $\overset{d}{=}$ represents equality in terms of distribution. Furthermore a time series can be second order self-similar, where the autocorrelations decay hyperbolically, if the following holds [19]

$$ACF(k) = 0.5\left[(k+1)^{2H} - 2k^{2H} + (k-1)^{2H}\right]\ ,\ 0.5 < H < 1 \tag{4.9}$$

In [19] Karagiannis et al. perform a comparison of the different available techniques for estimation of the Hurst exponent. The authors state that calculation of this exponent is not straightforward for two reasons. *The first* reason is because it can not be calculated definitively, but only estimated. *The second* reason is due to different methods producing different, sometimes conflicting results and its not clear which provides a better estimation. Hurst estimators can be classified into two main categories, the first one are estimators operating in the time domain and the second are estimators operating in the frequency domain.

- Time domain estimators - They investigate the power law relationship between a statistic in time series and the time aggregation of block size m. LRD exist if the statistic plotted versus m is a straight line in a log-log scale. The slope of the line is an estimate of the Hurst exponent. Types of estimators:
  - Rescaled range
  - Absolute values
  - Variance
  - Variance of residuals
- Frequency/Wavelet domain estimators - They examine if a time series' spectrum of energy follows the power-law behaviour. Types of estimators:
  - Periodogram
  - Whittle estimator
  - Abry-Veitch estimator

To simulate long range persistence the authors use two types of self-similar processes, fractional Gaussian noise (fGn) and Fractional Auto Regressive Integrated Moving Average (FARIMA). fGn is an increment of fractional Brownian motion (fBm), that is a random walk process with dependant increments. FARIMA(p,d,q) is a fractional version of auto-regressive moving average process ARMA(p,q). In these processes p represents the number of time steps in the past the present observation depends on, q represents the size of the moving average window and d in the FARIMA process represent the number of differences that will be applied on the original time series. For a process to be FARIMA d has to be non-integer and for the process to describe a LRD series 0<d<0.5, in which H=d + 0.5 [19].

Each of the estimators listed was tested against the two different types of long memory series. For each Hurst value between 0.5 and 1 with a step of 0.1, 100 fGn and 100 FARIMA series were generated. The authors recorded for each estimation technique the average estimated Hurst exponent over each of the 100 fGn and 100 FARIMA series. The authors of [19] found that there were significant variations in the estimated Hurst exponent between the different techniques. Frequency domain estimators appeared to be more accurate, with the Whittle and Periodogram estimators almost estimating exactly the Hurst exponents of the series generated through fGn. The Abry-Veitch estimator seemed to always over estimate the Hurst exponents, meanwhile time domain estimators failed to report the correct value. For time series generated through FARIMA the estimators were generally closer correct values, but none of the estimators managed to estimate the actual exponents.

To study the estimations sensitivity to the effect of various patterns common to time series such as periodicity, noise and trend, the authors of [19] generated the following time series and the estimators were tested again. In the following list the estimators together with the results obtained from the authors of [19] are given.

- **Cosine + white Gaussian noise** - Periodicity can mislead the Whittle, Periodogram, rescaled range and Abry-Veitch estimators. The estimation depends mainly on amplitude of the cosine function, with the estimations approaching 1 as the amplitude get larger
- **fGn series + white Gaussian noise** - Noise on LRD series caused all the estimators to underestimate the Hurst exponent, with the exception of Whittle and Abry-Veitch estimators. However depending on signal to noise ratio and Hurst exponents even these estimators can fail
- **fGn + cosine function** Periodicity on LRD data affected all the estimations. Depending on amplitude of the cosine, time series estimators underestimate Hurst exponent. Frequency besed methods tend to over estimate the Hurst exponent.
- **Trend** - The definition of LRD assumes stationary time series. Non-stationarity causes the Whittle estimate to be consistently 0.99, the Periodgram estimates Hurst exponent bigger than 1 and no other method produces statistically significant estimations.

This analysis indicates that such patterns significantly affect the estimations. No estimator seems to be consistently robust. However, signal processing techniques could be applied to overcome these limitations.

4.4. Effect of time scale on persistence

As it was mentioned, the Hurst exponent characterizes the global level of long range persistence on a time series. Under different time-scales it is expected that the morphology of the series will be affected [45]. In [45] Wang et al. have investigated the effect of different time scales using non-linear analysis. The authors of this paper have collected all the flights trajectories over the course of a week that pass over a particular waypoint. The arrival time of each flight to the waypoint was calculated and the time series of flights passing through this waypoint was created for three different time-scales 10,15 and 30 minutes. The result of this process is presented in Figure 4.4.

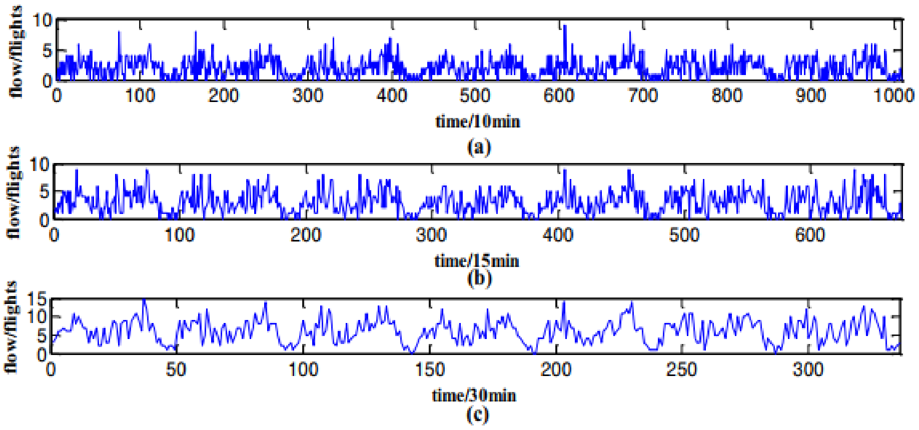


Figure 4.4: Air traffic flow time series at different time scales [45]

It is interesting to see that some of the peaks and valleys in the series occur simultaneously on the three different time scales. This occurrence is indication that traffic flows are self-similar in nature. Through using the rescaled range (R/S) method the authors of [45] have estimated the Hurst exponent for the three time series presented, their results are given in Table 4.1. Looking at this results it can be seen that as the scale gets bigger the Hurst exponent decreases, as such also the level of correlations in the data.

Time Scale	Hurst exponent
10 min	0.7224
15 min	0.6912
30 min	0.6440

Table 4.1: Results of estimating the Hurst exponent on different time scales [45]

In terms of making predictions it is desirable for the Hurst exponent to be as far away as possible from 0.5. The higher this value the higher the likelihood that the system will continue on a trend. The lower the value the higher the likelihood that the system will revert to the mean. As such, aggregating the time series data to different time scales and estimating the Hurst exponent could provide insights in the process of selecting the optimal look ahead time for making predictions.

As an alternative approach to time series aggregation the authors of [28] present a technique for applying a local variant of the Hurst exponent based on different time scales. Besides identifying the long-range persistence of the data the authors state that this technique can be used for identifying anomalies in the time series. This Time-Scale Local Hurst Exponent (TS-LHE) method is based on the idea of a moving window with variable size in which the (R/S) method is applied. In the next paragraphs the steps described by the authors of [28] to apply the TS-LHE on time series data will be stated. The process is also visualized in Figure 4.5.

1) Defining the range of moving window size W_s In similarity to the time scale aggregation of [45], the size of the moving window defines the time scales for which the correlations will be investigated. The minimum window size $W_{s,min}$ has to be no less than the minimum number of point for applying the R/S technique (4 points). The maximum window size $W_{s,max}$ for all practical purposes can be set to the number of observations available in the data N, in which case the global Hurst exponent of the data will be determined.

2) Padding the beginning and the end of the time series The end goal of the process is to have for each observation in the original series a value of the local Hurst exponent. To achieve this virtual observations of length $L = W_s/2$ are added to the beginning and the end of the original series, thus creating a new padded series of length $N + 2L$.

3) Applying for each window the R/S method For each window size and sample i in the original time series the TS-LHE is obtained, $H(i, W_s)$

4) Sliding the window Starting from the $j = L$ sample of the padded series, the moving window of size W_s is moved $j + 1$ after applying the R/S. The process is terminated when sample $j = N - L$ is reached. After this point the process is restarted with a new window size.

After the process has been repeated for all defined window sizes, the result is a matrix with as many rows as there are observations in the original series and as many columns as the window sizes tested. The matrix contains the TS-LHE for each sample and observation time scale. The process was tested on a set of synthetic seismograms. These seismograms are generated from a convolution of randomly distribuited spikes of varying amplitudes and a Ricker wavelet. In addition, a certain amount of white noise was added to investigate the effect of noise on the process.

The results of this are given in Figure 4.6. On the left the noise free seismogram is given and on the left a seismogram with a signal-to-noise ration of 10 dB. The red dots in Figure 4.6 indicate the locations of the random peaks some of which are heavily attenuated. The plot named CWT represents the continuous wavelet transform and is used as a comparison technique. Looking at the analysis of the noise free seismogram it can be seen that the peaks and valleys of the mean TS-LHE for the most part can be used to identify the locations of the spikes in the seismogram. Comparing the results of the TS-LHE with the CWT it can be seen that TS-LHE is not affected by the amplitude of the spikes, meanwhile the CWT is amplitude dependant. Looking at the seismogram with added noise, it can be seen that the CWT is more resistant to noise than the TS-LHE,

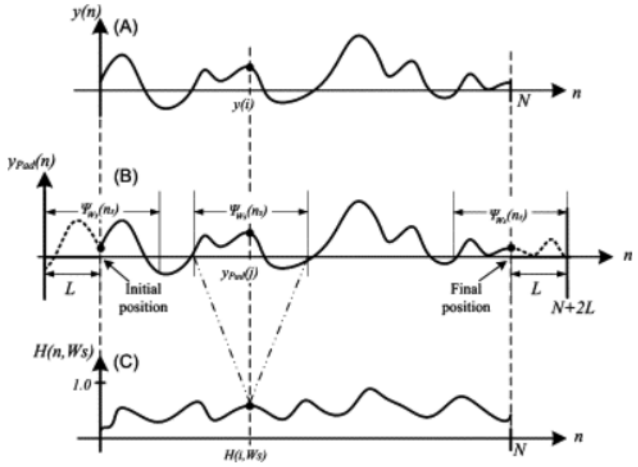


Figure 4.5: Process of applying the TS-LHE method proposed in [28]. Plot A is the original time series, plot B is the padded time series, plot C represents the result of the process for a certain window size.

however it is affected by the amplitude of the signal. It can be concluded that the TS-LHE technique can be used to identify changes in the series behaviour and the time scales of highest persistence in the series.

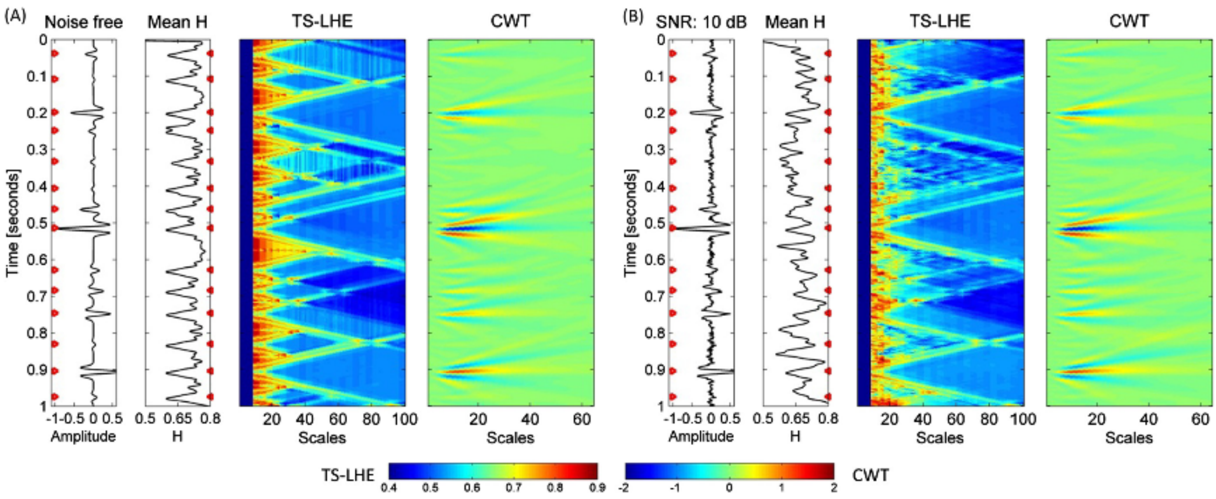


Figure 4.6: Results of applying the TS-LHE technique on synthetic seismograms in [28]

4.5. Recurrence plots

Recurrence analysis is a powerful method that can be used for analysis and identification of hidden patterns and dependencies in time series [22]. The most common tool to perform recurrence analysis is the recurrence plots. Such plots were used in [45] to study the stochasticity and determinism of air traffic flow time series with different time scales.

Creation of a recurrence plot starts with the reconstruction of the phase space. The phase space represents the set of states that the system can enter [37]. In [45] the authors used a method called the "C-C method" for reconstructing the phase space and in [22] the "Packard-Takens" procedure was used. Using the notation of [22], the reconstruction of the phase space leads to the following

$$F(t) = [x(t), x(t+\tau), ..., x(t+m\tau)] \tag{4.10}$$

where $F(t)$ is the m-dimensional pseudo phase space, $x(t)$ represents the state of the system at time t and τ is the delay period. A recurrence plot can be considered as a projection of this phase-space on a two dimensional plane. Consider two different points in the phase trajectory X_i and X_j, where i and j are related to time t. If the distance between these points is bellow a threshold ϵ, than point (i, j) on the recurrence plot is marked. The procedure is repeated for all pairs of points in the phase trajectory. Mathematically the procedure is formulated as follows

$$RP_{i,j} = \Theta(\epsilon - \|X_i - X_j\|) \tag{4.11}$$

where ϵ is the distance threshold, $\|X_i - X_j\|$ represents the distance and $\Theta(.)$ is the Heaviside function. An example of recurrence plots is given in Figure 4.7. In this figure the two line plots represent time series generated through Fractional Brownian Motion with Hurst exponents of 0,6 (top) and 0,9 (bottom). The left recurrence plot corresponds to the time series with H = 0,6 and the right to the series with H=0.9. Comparing the two recurrence plots it can be seen that for the most persistent series (right plot) the majority of points lie close to the diagonal, meanwhile as persistence goes down the points start to spread further away from the diagonal.

In [22] the authors present several metrics with which the topology of the recurrence plot can be analyzed to extract information on the series predictability. The first measure presented considers the density of points in a recurrence plot, the *recurrence rate* RR. The RR is defined as follows and it is probability that a state will re-occur. In [45] the authors state that the choice of ϵ should be such that the RR is greater than 1% and it also should be 15% less than the standard deviation of the pair wise distances in the phase trajectory.

$$RR = \frac{1}{N^2}\sum_{i,j=1}^{N} RP_{i,j} \tag{4.12}$$

The second measure is the probability that the system recurs to a neighbourhood of radius κ of a former point X_i after τ time steps, P_τ. It is formulated as follows

$$P_\tau = \frac{1}{N-\tau}\sum_{i,j=1}^{N-\tau} \Theta(\kappa - ||x_i - x_{i+\tau}||) \tag{4.13}$$

The next two measures are related to the presence of diagonal lines in the recurrence plot. The authors of [45] state that straight diagonal lines in a recurrence plot are an indicator of determinancy. Let $P(l)$ be the frequency distribution of lengths of diagonal lines. To quantify the amount determinancy the *measure of determinacy* DET, which determines the percentage of points on plot that fall on diagonal lines, is formulated as follows

$$DET = \frac{\sum_{l=l_{min}}^{N} lP(l)}{\sum_{i,j=1}^{N} R_{i,j}} \tag{4.14}$$

where l_{min} is the minimum length of a diagonal line. The second measure related to diagonal lines is the average length of diagonal lines and it is an indicator of the average time of predictability. It is formulated as follows

$$L = \frac{\sum_{l=l_{min}}^{N} lP(l)}{\sum_{l=l_{min}}^{N} P(l)} \tag{4.15}$$

The next two measures presented in [22] are related to vertical lines in the plot. In [45] the size and number of vertical lines are stated to be a measure of the stochasticity of the series. Let $P(v)$ be the frequency distribution of lengths of vertival lines and v the length of vertical lines. Similarly to DET but for vertical lines the *measure of laminarity* LAM is defined, as the percentage of plot points on vertical lines. It is stated in [22] that "LAM value characterizes the presence of fading states (i.e when the motion along the phase trajectory stops or moves very slowly)" and is formulated as follows

$$LAM = \frac{\sum_{v=v_{min}}^{N} vP(v)}{\sum_{i,j=1}^{N} R_{i,j}} \tag{4.16}$$

Similarly to the average length of diagonal lines L, the average length of vertical lines is also utilized. It is refered to as trapping-time TT and indicates the average time the system can spend in the neighbouhood of a particular state. It is defined bellow.

$$TT = \frac{\sum_{l=l_{min}}^{N} vP(v)}{\sum_{l=l_{min}}^{N} P(v)} \tag{4.17}$$

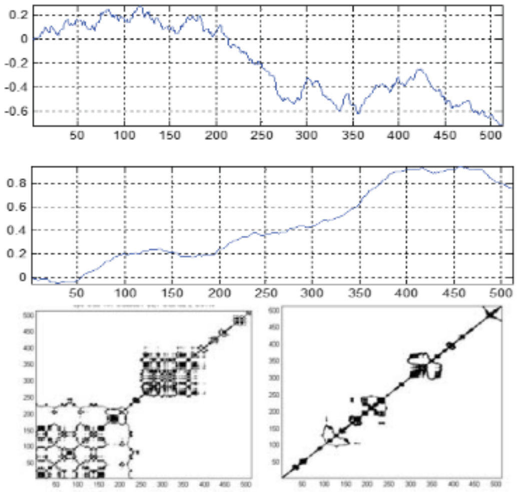


Figure 4.7: Time series and recurrence plot of Fractional Brownian Motion with Hurst exponent of 0,6 (top line plot and left recurrence plane) and 0,9 (bottom line plot and left recurrence plane). Obtained from [22]

From the discussion above and the measures presented in this section, it can be concluded that using recurrence plots can be a useful tool for exploratory analysis of the available data. Information such as average time of predictability can be used to determine the optimal prediction horizon and other measures such as LAM and DET maybe useful input in the prediction process.

4.6. Applications of non-linear features in machine learning

Paper 1 - Introduction

In [33] Qian and Rasheed hypothesize that forecasting time-series with Hurst exponent higher than 0.5 the forecasting error will be reduced. In order to perform this forecast they have opted to use a neural network. As input data they used the Dow-Jones index daily returns for the period between 2^{nd} of January 1930 to 14^{th} of May 2004. The Hurst exponent is calculated over periods of 1024 days. As training data 30 periods with large Hurst exponent and 30 periods with Hurst exponent close to 0.5 are selected.

Paper 1 - Methodology

Estimating the Hurst Exponent For estimation of the Hurst exponent, similarly to [45] and [28], the authors of [33] used the rescaled range method. For creation of the needed sub-periods they considered periods with lengths 16,32,64,...,1024 days, they note that this method does not perform well with periods less than 10 days. Furthermore it is noted that as a daily return $x(t)$ they utilize the log difference of the price $p(t)$, that is $x(t) = \log\big(p(t) - p(t-1)\big)$. The reason behind this choice is stated that it is a common method in the financial domain, however there is a deeper implication with using the difference. As it was seen in [19] when the Hurst exponent is to be estimated on a non-stationary time series all the methods tested failed to produce statistically significant results. Taking the difference of a non-stationary time series is a common transformation to convert the series to a stationary one [17].

Building confidence intervals As a final step the authors want to build the 95% confidence interval for the Hurst exponent of a time series representing a fully random time series. To achieve this they perform a Monte Carlo simulation where 10000 series of white noise with length of 1024 are generated. For each of the series the Hurst exponent is estimated and the results are averaged over all the estimated exponents. The process is repeated 10 times. The mean Hurst exponent at the end of the process was found to be 0.5454 and the mean standard deviation was 0.0485, thus the Hurst exponent can be anywhere in the range between 0.4503 and 0.6405 for the series to be a random series. As a result of this 30 periods with Hurst exponent bigger than 0.6405 and 30 periods with Hurst exponent between 0.54 and 0.55 were selected.

Data preparation In order to predict the next value in the series x_{i+1} given the time series $x_1, x_2, ..., x_i$, the authors have utilized Takens theorem to reconstruct the dynamical system. In this way a time-delay embedded vector $X_i = (x_i, x_{i+\tau}, x_{i+2\tau}, ..., x_{i+(d-1)\tau})$ is constructed, with d being the embedding dimension and τ the separation. Using the *auto-mutual information* method the separation was found to be 1 and through the *false nearest neighbours* method the embedding dimension was found to be from 3 to 5. The input X_i and the target x_{i+1} are both normalized to mean of 0 and standard deviation of 1. Finally to avoid over-fitting the dataset is split into train, validation and test sets.

Constructing the neural network The authors state that they did not observe an advantage of a deep architecture and for this reason only one hidden layer was used. For the learning algorithm the backpropagation with momentum, conjugate gradient method and Levenberg-Marquardt algorithms were tested, with the last being found to be the most effective. In order to select the number of hidden nodes the authors have utilized the following heuristic method

$$(N_{HN}+1)\cdot N_{HN} + (N_{HN}+1)\cdot N_{ON} = 1.5\sqrt{N_{data}} \tag{4.18}$$

with N_{HN} the number of hidden nodes, N_{ON} number of output nodes and N_{data} the length of the input series. So for a embedding dimension of 3 the above equation results in $(3+1)N_{HN} + (N_{HN}+1) = 1.5\sqrt{1024}$, which results in $N_{HN} = 10$. As a loss function the the authors have used the normalized root mean squared error (NRMSE). NRMSE varies between 0 and 1, being 1 when the predictions are close to the mean of the target outputs and 0 when all the predictions are correct.

Paper 1 - Results

After training and validation the authors ran the test set and recorded the NRMSE for each of 30 periods with Hurst exponent bigger than 0.65 and the 30 periods with Hurst exponent between 0.55 and 0.54. It was found for the periods with high Hurst exponent that the average NRMSE over the 30 periods was 0.9439 and for the other 30 periods the NRMSE was 0.9731. As a final step a students t-test was ran to test the null hypothesis of the mean values of the two sets of periods being equal. The t-statistic was found to be 7.369 and the p-value $7.029 \cdot 10^{-10}$, indicating that the null hypothesis can be rejected and the chance of equality being negligibly small. Thus the authors conclude that their initial hypothesis is valid, thus periods of high Hurst exponent can be forecasted with a lower error.

Paper 2 - Introduction

Non-linear dynamical features such as Hurst exponent, scaling exponent, approximate entropy and correlation dimension have been widely used in the health care industry for analysis and classification of EEG signals[4]. In [48] Yuan et al. are concerned with the problem of binary - classification of EEG signals. In particular they are interested to classify if a certain signal is ictal (having a seizure) or intericatal (for patients suffering from epilepsy this represents the nominal condition). As input features for the classification problem the authors have investigated three non-linear dynamic features of time-series, namely the Hurst exponent, the scaling exponent and approximate entropy. Furthermore they consider 3 different classification algorithms, extreme learning machine (ELM), a backpropagation network and a SVM.

Paper 2 - Methodology

Approximate entropy (ApEn) This is a statistical technique that is used to quantify the regularity of a time series. In particular this technique is focused in detecting the unpredictability of fluctuations in a time series [29]. Whereas the Hurst exponent is used in the identification of mean reverison, trending behaviour or complete randomness ApEn is aimed at identifying if a time series has repetitive patterns of fluctuation. As such it can be considered as a measure of predictability of a time series. ApEn is a non-negative number, the larger its value the more irregular the time-series is [48]. The methodology for its calculation is described in the appendix of [48].

Hurst exponent Without going much into detail as it has been already discussed several times, much like the other papers discussed in this chapter the authors of [48] have used the rescaled range method and its asymptotic behaviour to obtain this exponent.

Scaling exponent The scaling exponent α like the Hurst exponent is a measure of the long term correlations in a time-series. It is obtained through detrended fluctuation analysis (DFA), which is described in the appendix of [48]. As the name suggests the method is able to work with both stationary and non-stationary series as any existing trend will be removed. The value of the scaling exponent varies from 0 to 1.5 [32]. Similarly to the Hurst exponent when the scaling exponent is equal to 0.5 it represents a random walk[32]. When $0 < \alpha < 0.5$ the series has anti-persistence and when $0.5 < \alpha < 1$ the series has persistence. When $\alpha > 1$ the correlations exist but are not any more of the power-law type [32].

Machine learning The authors have investigated three different classification algorithms. The first two consist of a single hidden layer feedforward neural network and the third is a SVM. The two networks differ from each other in the learning algorithm used to tune the weights of the network. In the first case of an ELM the input weights and biases are selected randomly and the output weights in the end are solved for analytically. In the case of the backpropagation network all the parameters are tuned simultaneously and iteratively through a gradient descent. Each of the three non-linear dynamic features are used as seperate inputs to the three models to assess the performance of each of them individually and they are also combined in the end to investigate their performance jointly.

Paper 2 - Results

When using only the ApEn as input feature the authors obtained on all the three machine learning models a classification accuracy of 88%. The ELM required the least amount of time for training and testing followed by the backpropagation network and the SVM was the slowest. When using the Hurst exponent as a single input feature the classification accuracy was again 88% for all three models and the run times did not change from the prior case. When using the scaling exponent as input the accuracy of all three models degraded slightly with the extreme learning machine having an accuracy of 82% and the remaing two models resulted in an accuracy of 81.75%. Finally when combing Apen, Hurst and scaling exponent as input features the classification accuracy was found to be 96% for the ELM, 95.5% for the backpropagation network and 95.25% for the SVM.

Paper 3 - Introduction

In [15] Hatami et al. consider the task of time series classification. Incentivised by the classification performance achieved by convolutional neural nets (CNN) in image and speech classification the authors consider using such an architecture for classifying time series. In CNNs the input typically consists of an image which is passed through the different layers of the network and features of different hierarchical order are extracted from the image. In order to convert a time-series, which is a one dimensional vector, into an image the authors have converted such series to recurrence plots.

Paper 3 - Methodology

Recurrence plots The authors of the paper have utilized recurrence plots to encode time-series information into an image. As it was explained in section 4.5 the process involves reconstructing the phase space of the time series and checking for all pairs of states if the distance between them is less than a threshold. The authors of [15] note that through this thresholding and binarization of the resulting image some information is lost. To overcome this, creation of the recurrence plot has been modified to contain the distances between the pairs of states.

CNN for classification of time series In order to perform the classification the authors utilize two convolutional layers each of them followed by max pooling and dropout layers. The last two layers are employed for increasing the generalization ability of the network and preventing over-fitting. Through these two stages the most important features of the image are extracted and feature maps of different levels are created. Finally these feature maps are flattened and passed through a fully connected layer that leads to the output layer with as many nodes as there are classes to be predicted.

Input data & algorithms for comparison The time series that were used in this research where obtained from the UCR archive for time series classification. The series in this archive vary in terms of number of classes from 2 up to 60 classes for some series, in terms of training samples from 16 up to 8926 and length of time series from 24 up to 2709 observations. For the experiments 20 series from this archive were selected about a quarter of them are used for binary classification and the rest are multi-label classification problems.

The algorithms considered for comparison included 1NN-DTW (1 nearest neighbour dynamic time wrapping), Fast-Shapelets, Bag of Patterns, SAX-VSM (Symbolic Aggregate Approximation Vector Space Model). In addition to the above, three algorithms that involve transforming time series into an image are also used for comparison. For each of the series and algorithms considered the error rates were recorded and comparison is done on the basis of Number of Wins (Number of times an algorithm had the lowest error rate out of all considered for the dataset) and Average Rank (mean of the error rate ranking over all datasets)

Paper 3 - Results

After running all the algorithms and recording the error rates the results are presented in Table 1 of [15]. From this process it was found that proposed methodology yielded the best results both in terms of number of wins (10/20) and average rank (2.15). On 9 out of 20 tested datasets (6 binary, 3 multi-label classifications) the proposed solution scored an error rate of 0. On the datasets in which the method did not win the error rates varied between a minimum of 0.006 to a maximum of 0.29. The authors state that the proposed solution offers advantages in terms of recurrence plot being able to visualize certain patterns that are not easily seen otherwise and that the CNN is able to extract features of different levels from the time series.

5 Research definition

In the introduction it was concluded that one of the biggest bottlenecks that hinders the growth of the air travel within Europe is the ATM system and infrastructure. In particular a big part of this problem stems from ATFM. Given the lack of research aimed at predicting reactionary ATFCM regulations in Europe, this study is envisioned to fill this gap and obtain a better understanding of the application and evolution of ATFCM regulations.

In order to achieve this high level objective in the following section the main objectives and sub-objectives will be stated. Based on these objectives the main research question and related sub-questions will be formulated. Finally, in order to achieve the research objectives and give an answer to the research questions the research plan is presented.

5.1. Research objective

As it was seen in the literature study, research in the field of ATFM in the last years has been focused in prediction of air travel delays [6] or the most important factors that lead to a ground delay program [24]. In addition to that all of this research has been focused in the context of the US ATM system. For this reason the high level objective of this research is to create a model that will be able to predict the future ATFCM regulations in Europe. Thus the main objective of this study is:

"To construct a predictive model that, given a list of ATFCM notification messages (ANM) at time t is able to predict the characteristics such as, duration, time to activation and types of regulations at time $t + \Delta t$ by using historical ANMs to train a machine learning model for making predictions."

The second research objective is related to the prediction horizon Δt and is formulated as follows:

"To develop a methodology with which the maximum prediction horizon for an area control center can be quantified by using the concept of long-range dependence and in particular the Hurst exponent as an indicator of the predictability of the process."

5.2. Research question

Based on the research objectives that were defined in the section above the research questions have been formulated. With respect to the main objective the *main research question* is formulated as follows

" How can historical ATFCM notification messages be used in a machine learning model to predict for a certain prediction horizon the duration, time to activation and number of future ATFCM regulations during the tactical phase of operations for a certain area control centre?"

Following this main research question it can be broken down into the following sub-questions related to the prediction horizon:

1. How can one determine an appropriate value for the forecast horizon?
2. How can the value for this prediction horizon be verified?
3. How can the value for the prediction horizon be validated ?

Related to the machine learning context of the project the following sub-questions are formulated:

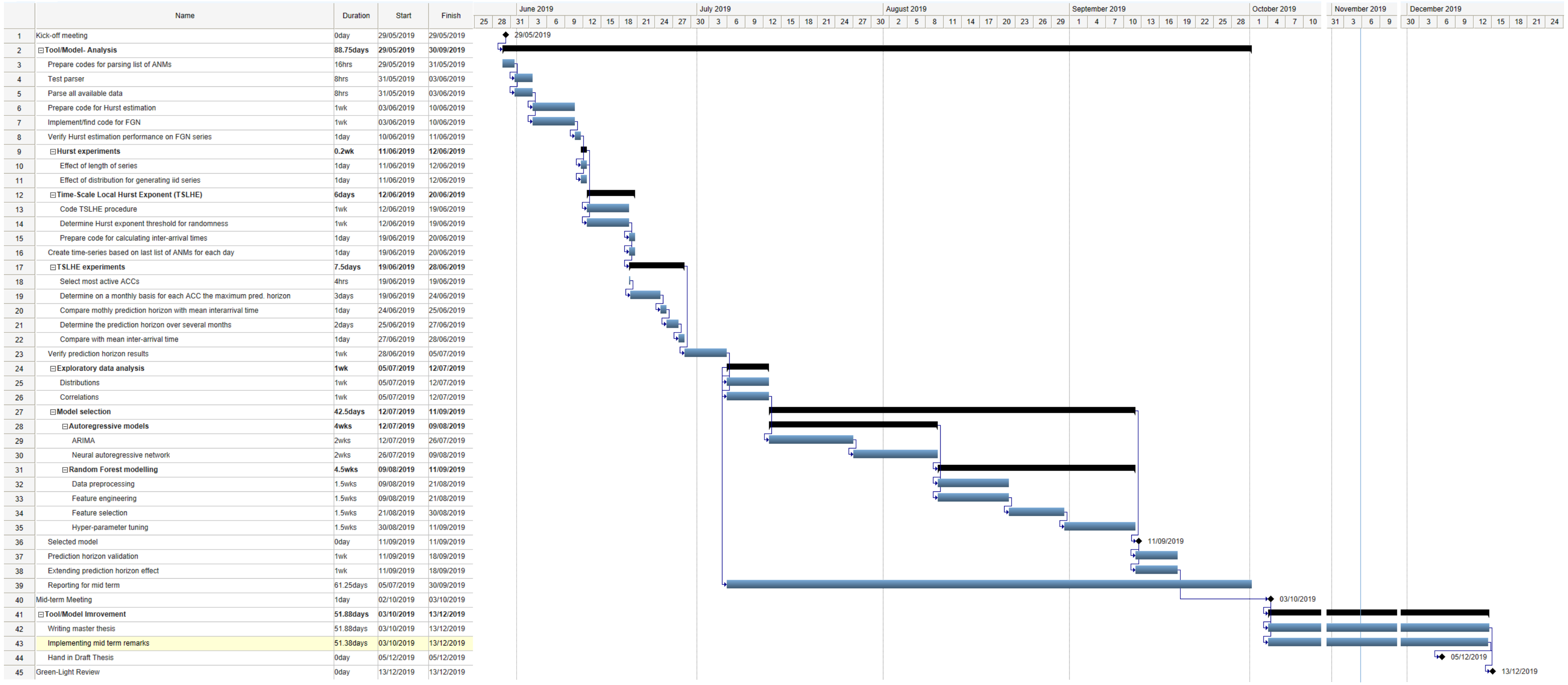
1. How to determine which features from the ATFCM notification messages are the most appropriate to be used as input variables to the machine learning model?
2. How to evaluate the predictive performance of the machine learning model?
3. How to analyze the impact of different prediction horizons on the output predictions?
4. How can the future predicted regulations be distinguished whether they are new, changed or cancelled regulations?

5.3. Research planning and management

In Figure 5.1 a Gantt chart detailing the long-term planning of the research is given. The activities planned have been based on the master thesis timeline of the Aerospace Engineering faculty. For the most part the task durations have been based also on this suggested timeline. The tasks shown in green represent an optimistic duration, so the shortest amount of time to be done. The orange tasks on the other hand represent a pessimistic duration. In this way upper and lower bounds for completion of the thesis are provided.

With regards to the tasks after the "Midterm review" their durations are assumed to be the same for both optimistic and pessimistic scenarios, only the start and end times deffer due to the dependencies on the initial tasks. In the scenario that the activities planned after handing in the literature review are completed in the soonest scenario, a two week break is planned such that the midterm review wont have to be on the first day of the new academic year. Besides that break the plan presented in Figure 5.1 it is assumed that the weekends are non-working days.

In terms of managing the research project, given its nature and the absence of a clear requirement space, it is deemed that a mix of agile project management methods together with long-term planning would be most suited. In this way the milestones and major deadlines are dictated by the long-term plan, while planning the activities to reach the end goal will be done on a bi-weekly basis.



	Name	Duration	Start	Finish
1	Kick-off meeting	0day	29/05/2019	29/05/2019
2	**⊟Tool/Model- Analysis**	**88.75days**	**29/05/2019**	**30/09/2019**
3	Prepare codes for parsing list of ANMs	16hrs	29/05/2019	31/05/2019
4	Test parser	8hrs	31/05/2019	03/06/2019
5	Parse all available data	8hrs	31/05/2019	03/06/2019
6	Prepare code for Hurst estimation	1wk	03/06/2019	10/06/2019
7	Implement/find code for FGN	1wk	03/06/2019	10/06/2019
8	Verify Hurst estimation performance on FGN series	1day	10/06/2019	11/06/2019
9	**⊟Hurst experiments**	**0.2wk**	**11/06/2019**	**12/06/2019**
10	Effect of length of series	1day	11/06/2019	12/06/2019
11	Effect of distribution for generating iid series	1day	11/06/2019	12/06/2019
12	**⊟Time-Scale Local Hurst Exponent (TSLHE)**	**6days**	**12/06/2019**	**20/06/2019**
13	Code TSLHE procedure	1wk	12/06/2019	19/06/2019
14	Determine Hurst exponent threshold for randomness	1wk	12/06/2019	19/06/2019
15	Prepare code for calculating inter-arrival times	1day	19/06/2019	20/06/2019
16	Create time-series based on last list of ANMs for each day	1day	19/06/2019	20/06/2019
17	**⊟TSLHE experiments**	**7.5days**	**19/06/2019**	**28/06/2019**
18	Select most active ACCs	4hrs	19/06/2019	19/06/2019
19	Determine on a monthly basis for each ACC the maximum pred. horizon	3days	19/06/2019	24/06/2019
20	Compare mothly prediction horizon with mean interarrival time	1day	24/06/2019	25/06/2019
21	Determine the prediction horizon over several months	2days	25/06/2019	27/06/2019
22	Compare with mean inter-arrival time	1day	27/06/2019	28/06/2019
23	Verify prediction horizon results	1wk	28/06/2019	05/07/2019
24	**⊟Exploratory data analysis**	**1wk**	**05/07/2019**	**12/07/2019**
25	Distributions	1wk	05/07/2019	12/07/2019
26	Correlations	1wk	05/07/2019	12/07/2019
27	**⊟Model selection**	**42.5days**	**12/07/2019**	**11/09/2019**
28	**⊟Autoregressive models**	**4wks**	**12/07/2019**	**09/08/2019**
29	ARIMA	2wks	12/07/2019	26/07/2019
30	Neural autoregressive network	2wks	26/07/2019	09/08/2019
31	**⊟Random Forest modelling**	**4.5wks**	**09/08/2019**	**11/09/2019**
32	Data preprocessing	1.5wks	09/08/2019	21/08/2019
33	Feature engineering	1.5wks	09/08/2019	21/08/2019
34	Feature selection	1.5wks	21/08/2019	30/08/2019
35	Hyper-parameter tuning	1.5wks	30/08/2019	11/09/2019
36	Selected model	0day	11/09/2019	11/09/2019
37	Prediction horizon validation	1wk	11/09/2019	18/09/2019
38	Extending prediction horizon effect	1wk	11/09/2019	18/09/2019
39	Reporting for mid term	61.25days	05/07/2019	30/09/2019
40	Mid-term Meeting	1day	02/10/2019	03/10/2019
41	**⊟Tool/Model Imrovement**	**51.88days**	**03/10/2019**	**13/12/2019**
42	Writing master thesis	51.88days	03/10/2019	13/12/2019
43	Implementing mid term remarks	51.38days	03/10/2019	13/12/2019
44	Hand in Draft Thesis	0day	05/12/2019	05/12/2019
45	Green-Light Review	0day	13/12/2019	13/12/2019

Figure 5.1: Gantt chart of the study phases and major milestones

Bibliography

[1] Airbus. Global Market Forecast, 2018. URL `https://www.airbus.com/content/dam/corporate-topics/publications/media-day/GMF-2018-2037.pdf`.

[2] James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

[3] Armin Bunde, Jan F. Eichner, Shlomo Havlin, and Jan W. Kantelhardt. Return intervals of rare events in records with long-term persistence. *Physica A: Statistical Mechanics and its Applications*, 342:308–314, 2004. doi: 10.1016/s0378-4371(04)00487-x.

[4] Roel F Ceballos and Fe F Largo. On the estimation of the hurst exponent using adjusted rescaled range analysis, detrended fluctuation analysis and variance time plot: A case of exponential distribution. *arXiv preprint arXiv:1805.08931*, 2018.

[5] Munevver Celik. NMOC Summer 18 Review, jan 2019. URL `https://www.eurocontrol.int/sites/default/files/events/presentation/nmuf2019_day1_1155_celik.pdf`.

[6] Alexander Estes, Michael O. Ball, and David J. Lovell. Predicting performance of ground delay programs. *Twelfth USA/Europe Air Traffic Management Research and Development Seminar*, 2017.

[7] Brian Flynn. ATFM in Europe, Oct 2013. URL `https://www.icao.int/APAC/Meetings/2013%20atfm%20sg2/09%20-%20ATFM%20in%20Europe.pdf`.

[8] Central Office for Delay Analysis. CODA DIGEST 2018. Technical report, EUROCONTROL, 2019.

[9] Thomas B Fowler. Heavy tails and the central limit theorem. *REVIEW*, page 25, 2007.

[10] Hardik Goel, Igor Melnyk, Nikunj C. Oza, Bryan Matthews, and Arindam Banerjee. Multivariate aviation time series modeling : Vars vs . lstms. 2016.

[11] Karthik Gopalakrishnan and Hamsa Balakrishnan. A comparative analysis of models for predicting delays in air traffic networks. *Twelfth USA/Europe Air Traffic Management Research and Development Seminar*, 2017.

[12] Karthik Gopalakrishnan, Hamsa Balakrishnan, and Richard Jordan. Stability of networked systems with switching topologies. In *2016 IEEE 55th Conference on Decision and Control (CDC)*, pages 2601–2608. IEEE, 2016.

[13] Asia/Pacific Air Traffic Flow Management Steering Group. *Asia/Pacific Framework for Collaborative Air Traffic Flow Management*. ICAO Asia and Pacific Office, 2017.

[14] Aurélien Géron. *Hands-On Machine Learning with Scikit-Learn & TensorFlow*. O'Reilly Media, 2017.

[15] Nima Hatami, Yann Gavet, and Johan Debayle. Classification of time-series images using deep convolutional neural networks. In *Tenth International Conference on Machine Vision (ICMV 2017)*, volume 10696, page 106960Y. International Society for Optics and Photonics, 2018.

[16] Harold E Hurst. The problem of long-term storage in reservoirs. *Hydrological Sciences Journal*, 1(3): 13–27, 1956.

[17] Rob J Hyndman and George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[18] ICAO. *Doc 4444 Procedures for Air Navigation Services: Air Traffic Management*. ICAO, 2016.

[19] Thomas Karagiannis, Mart Molle, and Michalis Faloutsos. Long range dependence- ten years of internet traffic modelling. *IEEE Internet Computing*, 8(5):57–64, 2004. doi: 10.1109/mic.2004.46.

[20] Sina Khanmohammadi, Salih Tutun, and Yunus Kucuk. A new multilevel input layer artificial neural network for predicting flight delays at JFK airport. *Procedia Computer Science*, 95:237–244, 2016.

[21] Young Jin Kim, Sun Choi, Simon Briceno, and Dimitris Mavris. A deep learning approach to flight delay prediction. *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*, 2016.

[22] Lyudmyla Kirichenko, Tamara Radivilova, and Vitalii Bulakh. Classification of fractal time series using recurrence plots. In *2018 International Scientific-Practical Conference Problems of Infocommunications. Science and Technology (PIC S&T)*, pages 719–724. IEEE, 2018.

[23] Giovanni Lenti. Understanding a difficult Summer 2018: why it happened?, Jan 2019. URL `https://www.eurocontrol.int/sites/default/files/events/presentation/nmuf2019_day1_1400_lenti_boydell.pdf?update`.

[24] Yi Liu and Mark Hansen. Predicting the initiation of a ground delay program. *Journal of Aerospace Operations*, 5(1):75–84, 2018.

[25] Network Manager. Network operations report 2018. Technical report, EUROCONTROL, 2019.

[26] Suvojit Manna, Sanket Biswas, Riyanka Kundu, Somnath Rakshit, Priti Gupta, and Subhas Barman. A statistical approach to predict flight delay using gradient boosted decision tree. *2017 International Conference on Computational Intelligence in Data Science(ICCIDS)*, 2017.

[27] Razvan Margauan. Introductory lecture to the Air Traffic Management, 2015.

[28] E Molino-Minero-Re, F García-Nocetti, and Héctor Benítez-Pérez. Application of a time-scale local hurst exponent analysis to time series. *Digital Signal Processing*, 37:92–99, 2015.

[29] George B. Moody. Approximate entropy (ApEn). URL `https://physionet.org/physiotools/ApEn/`.

[30] S. Niarchakou and M. Cech. *ATFCM Operations Manual.* EUROCONTROL, Brussels, Belgium, 2018.

[31] Daniel Alberto Pamplona, Li Weigang, Alexandre Gomes Barros, Elcio Hideiti Shiguemori, and Claudio Jorge Pinto Alves. Supervised neural network with multilevel input layers for predicting of air traffic delays. *2018 International Joint Conference on Neural Networks (IJCNN)*, 2018.

[32] C-K Peng, Shlomo Havlin, H Eugene Stanley, and Ary L Goldberger. Quantification of scaling exponents and crossover phenomena in nonstationary heartbeat time series. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 5(1):82–87, 1995.

[33] Bo Qian and Khaled Rasheed. Hurst exponent and financial market predictability. In *IASTED conference on Financial Engineering and Applications*, pages 203–209, 2004.

[34] Milton Raimundo and Jun Okamoto Jr. Application of Hurst Exponent (H) and the R/S analysis in the classification of forex securities. *International Journal of Modeling and Optimization*, 8:116–124, 04 2018. doi: 10.7763/IJMO.2018.V8.635.

[35] Gennady Samorodnitsky. Long range dependence. *Foundations and Trends® in Stochastic Systems*, 1 (3):163–257, 2007.

[36] MS Santhanam and Holger Kantz. Return interval distribution of extreme events and long-term memory. *Physical Review E*, 78(5):051113, 2008.

[37] T. D. Sauer. Attractor reconstruction. *Scholarpedia*, 1(10):1727, 2006. doi: 10.4249/scholarpedia.1727. revision #91017.

[38] Kamala Shetty, John Gulding, Hartmut Koelman, Mete Celiktin, and Rainer Koelle. Comparison of atfm practices and performance in the us and europe. In *2017 Integrated Communications, Navigation and Surveillance Conference (ICNS)*, pages 1C1–1. IEEE, 2017.

[39] EUROCONTROL Statistics and Forecast Service. European aviation in 2040 - challenges of growth, 2018.

[40] EUROCONTROL NMD/NOM/OPL Team. *European Route Network Improvement Plan-Part 3: Airspace Management Handbook.* EUROCONTROL, 2018.

[41] Balasubramanian Thiagarajan, Lakshminarasimhan Srinivasan, Aditya Vikram Sharma, Dinesh Sreekanthan, and Vineeth Vijayaraghavan. A machine learning approach for prediction of on-time performance of flights. *2017 IEEE/AIAA 36th Digital Avionics Systems Conference (DASC)*, 2017.

[42] Umberto Triacca, Antonello Pasini, and Alessandro Attanasio. Measuring persistence in time series of temperature anomalies. *Theoretical and applied climatology*, 118(3):491–495, 2014.

[43] EUROCONTROL Operational Analysis Reporting Unit. Network Operations Report – March 2019, 2019.

[44] J Eduardo Vera-Valdés. On long memory origins and forecast horizons. *arXiv preprint arXiv:1712.08057*, 2017.

[45] Chao Wang, Zhaoyue Zhang, and Ming Zhu. Nonlinear dynamic analysis of air traffic flow at different temporal scales: Nonlinear analysis approach versus complex networks approach. *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, 2018. doi: 10.1109/qrs-c.2018.00079.

[46] F Wang, P Weber, K Yamasaki, S Havlin, and HE Stanley. Statistical regularities in the return intervals of volatility. *The European physical Journal B*, 55(2):123–133, 2007.

[47] Jiang Xu-Rui, Wu Ming-Gong, Wen Xiang-Xi, and Wang Ze-Kun. Application of ensemble learning algorithm in aircraft probabilistic conflict detection of free flight. *2018 International Conference on Artificial Intelligence and Big Data (ICAIBD)*, 2018.

[48] Qi Yuan, Weidong Zhou, Shufang Li, and Dongmei Cai. Epileptic EEG classification based on extreme learning machine and nonlinear features. *Epilepsy research*, 96(1-2):29–38, 2011.