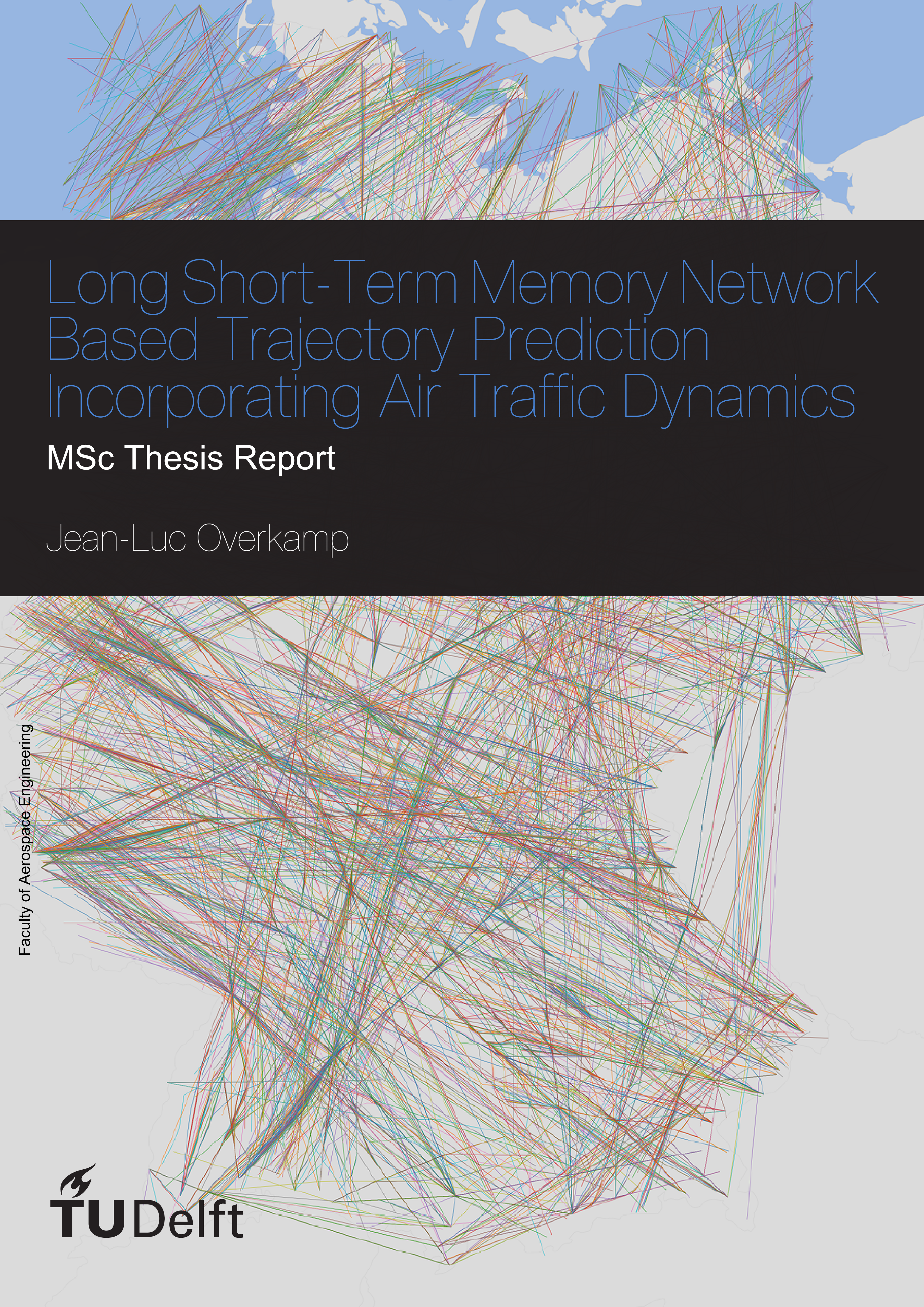


Long Short-Term Memory Network Based Trajectory Prediction Incorporating Air Traffic Dynamics

MSc Thesis Report

Jean-Luc Overkamp

Faculty of Aerospace Engineering

Long Short-Term Memory Network Based Trajectory Prediction Incorporating Air Traffic Dynamics

MSc Thesis Report

by

J.L. Overkamp

to obtain the degree of Master of Science
at the Delft University of Technology,

Student number:	4350170	
Date:	September, 2021	
Supervisors:	Dr. ir. J. Sun,	TU Delft, daily supervisor
	Prof. dr. ir. J. M. Hoekstra,	TU Delft, supervisor

TU Delft

Preface

The document in front of you is the final product of a Master of Science thesis project. The thesis is conducted with the Communication, Navigation, Surveillance/Air Traffic Management group, part of the Control and Simulation department at the Faculty of Aerospace Engineering at the Delft University of Technology. This MSc thesis report includes a scientific paper on the primary findings of this study as well as the preliminary report, summarizing the first half of this thesis project.

I would like to express my gratitude to my supervisors Junzi Sun and Jacco Hoekstra. Your extensive knowledge on the topic, critical questions, and many discussions challenged me to stay sharp, allowed me to think like an engineer, and helped improve the quality of my work. Working independently from home was not always easy but your time and flexibility gave me the motivation needed to complete this thesis, of which I am -and hopefully you are- proud of.

I would like to thank Tim, Jelle, Gervase, and Lars for proving feedback on the report. Romi, thank you the endless support and for sitting next to me in the home-office for so many days. To my friends and especially family, I would like to thank you for laying the foundation to make this possible and supporting me in completing my studies.

J.L. Overkamp
Delft, September 2021

Contents

I

Scientific Paper

Long Short-Term Memory Network Based Trajectory Prediction Incorporating Air Traffic Dynamics

Jean-Luc Overkamp,
supervised by Junzi Sun and Jacco M. Hoekstra
Control and Simulation, Faculty of Aerospace Engineering, Delft University of Technology

***Abstract*—Accurate four dimensional trajectory predictions are required for the continued implementation of Trajectory Based Operations. In addition, decentralized, free routing can make medium- to long-term flight trajectories more difficult to predict. Novel trajectory prediction techniques are needed, independent of waypoint-to-waypoint navigation and air traffic control operator behavior. The effect of air traffic dynamics on flight trajectories are currently under-explored. This research aims to improve the accuracy of medium- to long-term 4D flight trajectory predictions by incorporating a model that encompasses the dynamics of the air traffic situation. Data-driven techniques are well-suited to trajectory prediction purposes as high-fidelity air traffic and environmental data are more widely available. A statistical analysis is first conducted to select the most suitable air traffic dynamics features for the purpose of trajectory prediction. Six features are identified that correlate to the track deviation. This paper proposes a composite, deep neural network to predict individual trajectories, merging a *Long Short-Term Memory* (LSTM) network with a *2D convolutional LSTM* (ConvLSTM) based network. The air traffic dynamics features are translated to a spatiotemporal map and processed by a series of ConvLSTM layers. 75% of the 4D flight trajectory predictions, with traffic density as a feature, have a cross-track error under 15 NM at a 28 minutes of look-ahead time. Moreover, under certain conditions, the proposed model is able to make trajectory predictions with higher accuracy compared to the filed flight paths. However, there is no observable improvement in the trajectory prediction accuracy by incorporating air traffic dynamics into the proposed composite ConvLSTM model compared to a stacked LSTM model without the air traffic dynamics. The surrounding air traffic dynamics are of some, but minimal, influence to the 4D trajectories of individual flights. With an improved model, it is likely that traffic density can contribute to more accurate 4D flight trajectory predictions. The results encourage further research on the effects of air traffic dynamics on the individual trajectories.**

Index Terms*—Air traffic dynamics, air traffic complexity, trajectory prediction, 4D trajectories, Long Short-Term Memory networks, ConvLSTM.

I. Introduction

THE operational capabilities of traditional *Air Traffic Management* (ATM) are reaching the ceiling of capacity, efficiency, and cost effectiveness. *Trajectory Based Operations* (TBO) are part of the solution and allow increased capacity, safety, and efficiency. The adaptation of TBO demands an improvement in *Four Dimensional* (4D) trajectory prediction accuracy because of the interconnected nature of all stages of the trajectory cycle. In addition to TBO, the implementation of free routing provides a pilot more freedom to fly the preferred flight path. However, losing the dependency on waypoint-to-waypoint navigation, (inefficient but predictable) *Air Traffic Control* (ATC) interventions, and use of standardized routes reduces the predictability of flight trajectories.

In a controller- and route independent environment, it is expected that the air traffic situation encountered en-route affects the flight intent of individual aircraft, thus changing the flown flight path. Consequently, a relationship between air traffic dynamics and individual en-route flight paths can be used to predict trajectories. However, the effects between air traffic dynamics and the flown trajectory and how this can be used to predict flight trajectories has not been previously studied. The combination of recently published high-fidelity air traffic and environmental data with data-driven techniques provide a novel opportunity to address this research gap.

The objective of this research is to improve the accuracy of medium- to long-term flight trajectory predictions by incorporating a model that encompasses the dynamics of the air traffic situation. To achieve this, a two-step method is conducted. In the first phase, a statistical analysis is conducted between the air traffic dynamics and individual trajectories. The purpose of the first phase is to understand, identify, and select relevant features of the air traffic dynamics that have a relationship with individual trajectories. In the second phase, the trajectory prediction is performed. The purpose of this phase is to design and test a novel data-driven trajectory predictor that incorporates the selected air traffic dynamics features, thereby aiming to improve the accuracy of the trajectory prediction.

This research paper presents the related works (Section II); followed by the methodology (Section III), including the calculation of the air traffic dynamics features, the statistical analysis, and the deep learning trajectory prediction model; in Section IV, the experimental set-up of the trajectory prediction phase is presented; the results of the statistical analysis are presented in Section V; followed by the results of the trajectory prediction phase, in Section VI; last, the discussion, conclusion, and recommendations are provided in Section VII, VIII, and IX, respectively.

Junzi Sun is an assistant professor of Communication, Navigation, Surveillance/Air Traffic Management with the Control and Simulation research group at the TU Delft

Jacco M. Hoekstra is a full professor of Communication, Navigation, Surveillance/Air Traffic Management with the Control and Simulation research group at the TU Delft

II. Related Works

A. Developments in Air Traffic Management

The Single European Sky initiative and its USA equivalent, NextGen, are ongoing modernization projects aiming to increase capacity, overall efficiency, and safety, as well as to reduce environmental impact. A cornerstone of both projects is the ATM concept of TBO[1].

TBO places 4D trajectory information in the center of the ATM chain, demanding that all stages of the trajectory life-cycle -from planning to execution and amendments- are linked. In TBO, the pilot and airliner have the freedom to choose and optimize the route, and are not strictly limited to waypoint-to-waypoint navigation upon ATM instruction. Interconnection, sharing of information, and the integration with decision support tools will allow optimized services for all ATM stakeholders. This integration, and thus dependability, of services can only be attained with high accuracy (4D) trajectory prediction. Moreover, ATC capacity, procedures, and a lack of shared information place significant restrictions on the demand and capacity balance. Accurate global navigation and improved decision support tools for ATC have allowed an increased shift from ground navigational aid use to area navigation operations, providing timely alerts when an aircraft deviats from its assigned route. Naturally, this also places an increased requirement on the accuracy to be able to predict the position of an aircraft.

Implementation of the *Free Route Airspace*[2] (FRA) is a key element of TBO. FRAs are specified volumes of upper airspace that support the concept of operations in which a user has the freedom to plan a route between defined entry and exit waypoints. FRA has been implemented in three quarters of European (upper) airspace. Flights remain subject to ATC and, depending on airspace availability, routing is possible via intermediate waypoints. The freedom to execute any preferred route by pilots reduces the predictability because of reduced ATC standardization and reduced dependency on (consistent, thus predictable) controller strategies. Next generation 4D trajectory predictors therefore need to be less controller and route dependent and more dependent on individual aircraft behavior.

B. Air Traffic Dynamics

Literature related to the modelling of air traffic dynamics can be roughly divided into two categories, depending on the phase of flight: (pre-)tactical or in-flight. The tactical flight planning phase refers to hours or minutes prior to departure. In this phase, the Air Traffic Flow and Capacity Management requires air traffic dynamical models to predict and solve the demand and capacity balance. Currently, the in-flight phase requires an air traffic dynamics model to model the real-time, local, air traffic complexity. Quantifying the sector complexity can help the Air Navigation Service Provider predict the workload of controllers and make tactical decision on how to 'simplify' the traffic distribution to relieve the *Air Traffic Control Operator* (ATCO) workload. In fact, it is the so-called flight intent of pilots and ATCOs that causes track deviations from planned trajectories, not the demand and capacity balance. For these reasons, this literature overview will focus on air traffic complexity and workload modeling in order to quantify the air traffic dynamics.

1) Air Traffic Complexity and Workload Modelling: A quantitative assessment of the cognitive complexity was needed to control and understand sector capacity as well as to make progress in the automation of ATM. However, the subjective ATCO workload is highly complex and includes qualitative as well as quantitative metrics. The relationship between complexity and workload is mediated by several factors, including equipment quality, individual controller differences, controller cognitive strategies, and ATC procedures [1], [2], [3]. Multiple efforts have been taken to objectively measure and model the air traffic complexity, which will serve as a basis for the air traffic dynamical models.

NASA was first to propose *Dynamic Density* (DD) [4]. This metric is composed of traffic density and traffic complexity, which constitutes of various weighted complexity factors. The purpose was to measure the ATCO workload. It is a weighted linear function, initially including eight air traffic complexity terms in addition to air traffic density. Multiple variations of this model have been validated, contributing to a vast set of air traffic complexity factors. A more novel approach to modelling controller workload, but with an effectively similar working principle to the linear regression of DD, is the use of *Neural Networks* (NNs). In [3] and [5] a large variety of complexity factors is fed into a standard multilayer perceptron. The network outputs an estimate of the complexity. Using such methods reduces the dependency on the concept of operations, making it a better universal approximator.

2) Next Generation Complexity Modelling: With the increase of aircraft on-board autonomy and self-separation, a renewed push is made to model the air traffic dynamics and complexity without the controller workload in the loop. This will lead towards a partially decentralized control scheme for ATM. Piroddi and Prandini argue that the next generation ATM complexity evaluation will support the functionality of on-board trajectory prediction [6], [7].

Three-dimensional aircraft proximity maps evaluate the future probability of presence of aircraft at any given point of the airspace in [8]. Similarly to these proximity maps, Prandini introduces a conflict map to study the effect of these probabilities on surrounding aircraft [9]. This concept is further developed in [10], where the uncertainty in future aircraft positions are taken into account to evaluate the complexity. This approach is potentially useful as it includes timely identification of conflict situations which require maneuvering.

On a very different note, in [1] it is questioned if the complexity factors that are applied in DD models during en-route based operations are also applicable to TBO. This led to a number of factors that have been shown to accurately represent ATC complexity under TBO conditions.

In a geometric approach, complexity is defined as a geometric zone of influence, which is an envelope of possible

[1] https://www.icao.int/airnavigation/tbo/Pages/Why-Global-TBO-Concept.aspx

[2] https://www.eurocontrol.int/concept/free-route-airspace

motions that lead to the set of possible locations reachable by an aircraft from its intended trajectory. This approach has been researched in various studies, including in the Solution Space Diagram research [11]. To reduce the subjectivity error, Delahaye et al. [12] evolved this approach and evaluate complexity in which the relative motion of aircraft represents intrinsic traffic disorder. The traffic disorder is represented by a three-dimensional complexity coordinate, composed of three axes: density, divergence/convergence, and (in)sensitivity. The spatial approaching rates have also been successfully explored in [13] as a measure for complexity. Perhaps the most significant work on modelling the intrinsic complexity of the air traffic is also by Delahaye et al. [14], by pure dynamical systems modeling. The so-called Lyapunov exponents are a traffic disorder metric that measure the sensitivity to initial conditions of the dynamical system.

C. Data Driven Trajectory Prediction

Three classes of trajectory predictors are observed: nominal (deterministic), worst-case, and probabilistic. Nominal methods predict a single trajectory by propagating the observed aircraft and atmospheric states along a single trajectory. Generally speaking, this means that the models used to calculate a trajectory are (analytical) performance based models. Data-driven prediction models are inherently probabilistic because historical data or input data with a specific distribution are translated into the most-likely numerical solution. Data-driven techniques allow trajectory predictions based on machine learning and agent-based modeling methods, considering all relevant, actual historical data, including contextual features. Therefore, these methods can encapsulate more parameters and underlying relations which remain hidden with analytical methods. SESAR's *Data-driven AiRcraft Trajectory prediction research*[3] (DART) project emphasizes the importance and potential benefits of such novel applications and has conducted several comparative studies [15].

An important remark is that many studies apply clustering of the highly repetitive flight trajectories and thus resemble a classification problem. Clustering, often applied in conjunction with data driven trajectory predictions, is not applied in this research because the application of this research is concerned with FRA and en-route flights, aiming to predict trajectories independent of standard routes. The trajectory prediction problem in this research is therefore a continuous time-series regression problem without primarily discrete nor categorical elements.

In 2013, Leege et al. [16] proposes *Generalized Linear Model* (GLM) to predict the time over points along the fixed arrival route. A novelty is the stepwise regression approach to determine the explanatory power of each input variable. However, by direct comparison between *Multiple Linear Regression* (MLR) and NNs, the NNs outperform MLR for the trajectory prediction task [17], [18]. Ayhan [19] proposes a *Hidden Markov Model* (HMM) to predict 4D aircraft trajectories, taking into account atmospheric uncertainty. The airspace is represented as set of (discrete) 3D cubes, where each cube contains homogeneous environmental data. A trajectory is then predicted by a sequence of these cubes with spatiotemporal attributes. This successful concept can be translated to this research, in which the cubes contain air traffic dynamics instead of weather information.

Next, *Recurrent Neural Networks* (RNNs) are especially suitable for sequence modeling and employing (time-varying) spatiotemporal patterns, as observed with (multiple) aircraft trajectories. The LSTM network is a popular adaptation of a RNN and has been proposed and applied to trajectory prediction [20], [21]. For certain time-series prediction tasks, it has been shown that LSTM NNs outperform the autoregressive moving average and support vector regression models [22], [23]. The LSTM has been shown to also outperform Markov Models for flight trajectory prediction [24]. Applied ensemble methods such as *Gradient Boosting Machines* (GBM) and random forest models, are well-suited to the trajectory prediction problem [25], [26]. In direct comparison, GBM and random forest models perform similarly and outperform other ensemble techniques as well as logistic regression methods [27], [28].

A novel extension of a LSTM layer is the *2D Convolutional LSTM* (ConvLSTM) layer, which allows spatiotemporal sequence forecasting. A ConvLSTM network combines pattern recognition of 2D feature maps of convolutional networks with the memory properties of a LSTM network (needed for time-series). Novel and successful applications include precipitation nowcasting[4] [29] and weather forecasting[5]. In 2019, a LSTM NN was proposed for trajectory prediction that embeds convolutional layers to incorporate the convective weather condition along with a flight plan [30]. This is a similar approach to this research, except the weather condition is substituted with air traffic dynamics. Reference [30] trains a model based on a single trajectory. The ConvLSTM layers provide an opportunity to include spatiotemporal air traffic dynamics feature maps into a deep LSTM network and extract complex patterns that are useful for trajectory prediction.

A deep LSTM network with ConvLSTM layers is expected to be the most suitable data-driven trajectory predictor for the application of this research because of the absence of clustering; the sequential, time-varying nature of flight trajectories; and the spatiotemporal features of air traffic dynamics.

III. Methodology

The methodology proposed in this section consists of four parts. First, the chosen input data is presented. Second, the air traffic dynamics model is described. Third, the method of the statistical analysis of these air traffic dynamics is proposed. Then, the fourth section explains the working principle of the LSTM and the ConvLSTM layers; discusses the data preparation needed to perform the trajectory prediction; and last, proposes the deep learning model design.

[3] http://dart-research.eu/

[4] Nowcasting is forecasting of the near-future.

[5] https://medium.com/@rajin250/precipitation-prediction-using-convlstm-deep-neural-network-b9e9b617b436

A. Input Data

The data used for this study includes flight points, flight airspaces, and flight details sourced from the Eurocontrol R&D Data Archive[6]. This data is collected from all commercial flights operated in and over Europe. It is a processed data set to ensure accuracy that includes data from air navigation service providers' flight data systems, radar, and data link communications. Each flight is recorded as a sequence of flight points with time, flight level, latitude, and longitude at each point. Moreover, the data set includes key details on the flight, such as departure and arrival airport; departure time; and aircraft type and operator.

The Eurocontrol September 2018 Monthly Network Operations Report reveals that from the available months and years of the Eurocontrol R&D data set, September 2018 had the highest ever recorded daily traffic levels at around 34,000 flights [31]. The region that experienced the highest level of en-route delay in September 2018 is the Karlsruhe *Upper Area Control* (UAC). It is expected to be more likely for a flight to deviate from the filed flight path when passing through a highly congested region, thus revealing stronger patterns between air traffic dynamics and flown trajectories. At night, the airspace is not congested and so a variety of air traffic dynamics is included in the selected data set. In line with the objective of this study, Karlsruhe UAC is a FRA enabled area. It is for these reasons that the Karlsruhe UAC is a suitable region to apply this research, although the length of some flights extend outside the UAC.

B. Air Traffic Dynamics

The air traffic dynamics are modeled by selecting features which capture the most mathematically fundamental and intuitive relationships which might be considered by ATCOs and pilots in-flight. The reason to choose features with distinctive and intuitive differences is so that this can in turn lead to distinctive, understandable, and reproducible differences in the trajectory prediction results. The proposed features that constitute the model are based on the DD model, the TBO complexity indicators, and one feature from the Dynamic Weighted Network. The proposed features are shown to correlate with (subjective) complexity in various studies [1], [3]-[5], [13]. The selection is made based on the literature study, evaluating each feature based on the following: controller independence, route independence, objectivity, and expected indicator of complexity. The following list includes the proposed air traffic dynamics features, where each feature is computed for each grid at each time step :

1) **Traffic Density** (*TD*);
2) **Aircraft count** (*ACC*);
3) **Heading change** Count of aircraft making > 15°heading angle change within 2 minute period (*HdgCnt*) and average heading change for aircraft passing this threshold (*HdgAvg*);
4) **Speed change** Fraction of aircraft with an airspeed change of >10 kts within a 2 minute period (*SpdCnt*) and average airspeed change for aircraft passing this threshold (*SpdAvg*);
5) **Altitude change** Fraction of aircraft making >750 ft altitude change within 2 minute period (*AltCnt*) and average heading change for aircraft passing this threshold (*AltAvg*);
6) **Minimum distance** Count of aircraft pairs at 3D Euclidean distance less than 5 NM (*MinDst5*), 5-10 NM (*MinDst10*), and 10-50 NM (*MinDst50*) separation.;
7) **Mean separation** Average horizontal distance between all aircraft (*AvgDst*);
8) **Heading Variance** Standard deviation of aircraft heading angles (*SDHdg*);

The purpose is to study and utilize the effects of the air traffic dynamics on a single trajectory, therefore the air traffic dynamics features must represent the interaction effects between surrounding traffic, but not the entire sector. For instance, a pilot will not alter the flight path when four aircraft at a distance of 200 NM are changing their heading angle. Computing each feature in a grid representation allows for the consideration of surrounding air traffic in computing air traffic dynamics features, as seen in Figure 1. To capture possible differences in the consideration horizon of pilots and ATCOs alike, the analysis is repeated for varying grid sizes: 0.5°, 1°and 1.5°. A 1°x1°grid is in the order 40x60 NM, but slightly varies depending on the latitude. This approach has previously been used for trajectory prediction based on weather information in various papers [19], [28]. *Kernel Density Estimation* (KDE) is applied to mitigate the discretization error of some features.

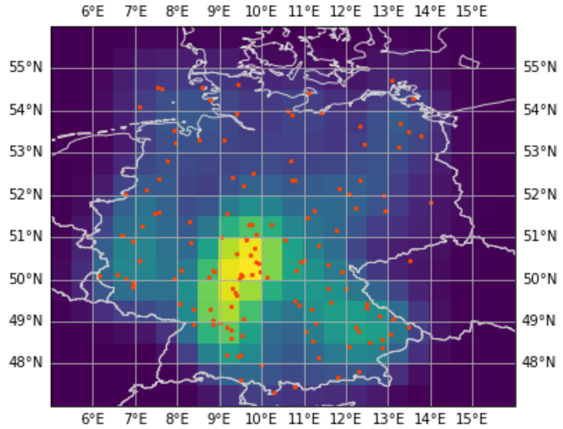


Fig. 1: The computed KDE of Traffic Density, shown on the grid representation with 0.5 degree grid size.

C. Statistical Analysis

The purpose of the statistical analysis is to provide a basis for the trajectory prediction by better understanding the air traffic dynamical behaviour and flight response to certain features. The analysis also helps to indicate the potential predictive power of the selected features. This in-turn allows to verify the feature inputs to the prediction model, aiming to increase the prediction model performance.

[6] https://www.eurocontrol.int/dashboard/rnd-data-archive

The statistical analysis is conducted between the dependent variables as listed in Section III-B and three independent variables: delay, instantaneous track deviation, and aggregated track deviation. The three independent variables are metrics used to evaluate the trajectory. In this paper, delay is defined as the difference between the actual and filed duration of time that an aircraft takes to pass through a sector. This is done by comparing the filed and actual entry and exit times for each flight. This way, the delay of a flight prior to entry does not affect the results. The purpose of this metric is characterize the effect of the air traffic dynamics features on the temporal behavior of flights. Track deviation is defined as the geodesic distance between the actual flight point and the nearest filed flight point. The vertical element of track deviation is a direct comparison between the altitude of the filed and the nearest actual flight point: *Flight Level* (FL) deviation. The instantaneous track deviation equals to the average track deviation of all flights in each grid at each time step. The purpose of this metric is to capture generalized spatial effects that are time-invariant. In other words, do the local air traffic dynamics of surrounding traffic have a direct (short-term) effect on the flown trajectory? The aggregated track deviation is calculated through time integration of the track deviation between the first and last point on the trajectory for each flight. The purpose of this metric is to capture possible delayed or anticipated effects to the air traffic dynamics features.

The instantaneous track and FL deviations are tested against the corresponding air traffic feature grid values, at each time step. The instantaneous features are the 13 features from Section III-B plus the KDE of the following features: *HdgCnt*, *AltCnt*, *SpdCnt*, *MinDst5*, *MinDst10*, *MinDst50*.

The aggregated track deviation and delay -one per flight- are tested against the cumulative air traffic feature grid values encountered by all flight points per flight. The cumulative features consists of the 13 features, where only *TD* is calculated as a KDE, plus the size of each trajectory, measured by the amount of (evenly sampled) flight points.

The statistical analysis is conducted by performing Spearman's coefficient of rank correlation because the data is numerical and does not pass the tests for normality. The Spearman coefficient is a Pearson's coefficient between the rank variables and given in Equation (1)[7]. The Spearman's coefficient of rank correlation is a non-parametric statistical test and provides a measure of how close two sets of rankings correlate with another. The threshold for rejecting the null-hypothesis lies at p < α = 0.02. This threshold is often set at 0.05, but the amount of data samples is large, so a higher threshold can be demanded.

$$\rho = 1 - \frac{\text{cov}(R_x, R_y)}{\sigma_x, \sigma_y} = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \tag{1}$$

Where, $R_{x,y}$ denotes the rank variables, $\sigma_{x,y}$ is the standard deviation, d_i is the difference in paired ranks, and n is the number of pairs. Moreover, cross-correlation testing is performed to detect any pair of features for which the distributions are too similar. Cross-correlation testing is a step in the statistical analysis process for dimensionality reduction and reducing bias. The cross-correlation is tested by computing the normalized covariance matrix.

D. Trajectory Prediction

In the second phase of this research, a LSTM based deep NN with two branches is proposed that can incorporate air traffic dynamics features to predict 4D trajectories. The flight point and information input data is passed to one branch with a LSTM layer. The spatiotemporal air traffic dynamics maps serve as inputs to the branch with ConvLSTM layers.

1) LSTM Network: The LSTM is a type of RNN with a specialized LSTM module designed to handle long-term dependencies [32]. Instead of a single RNN module with a *tanh* activation function, the LSTM has a unique module that adds and removes information to the cell state through structures called gates, in order to only pass on relevant information. This avoids the vanishing gradient problem as seen with RNNs. The structure of the repeated LSTM module is shown in Figure 2.

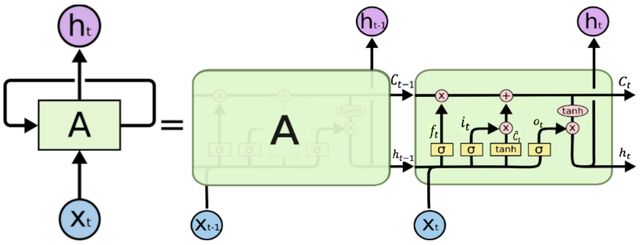


Fig. 2: An unrolled RNN with the repeated module showing the LSMT with four interacting layers[33].

Equation (2) gives the gates, cell state, and activation function equations, as explained below.

$$\begin{cases} f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + b_f\right) \\ i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + b_i\right) \\ \tilde{C}_t = \tanh\left(W_{xC}x_t + W_{hC}h_{t-1} + b_C\right) \\ C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \\ o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + b_o\right) \\ h_t = o_t \odot \tanh\left(C_t\right) \end{cases} \tag{2}$$

Where $\odot$ represents an element wise product. There are three gates: the forget, f_t; the input, i_t; and the output gate, o_t. Each gate contains a sigmoid (σ) layer which passes through either all the information (1) or none (0). The forget gate, f_t, can store or disregard the information from the cell state at $t-1$: C_{t-1}. The input gate, i_t, controls the information that will be added to the cell state, C_t. The input gate first decides to update the cell state or not through the sigmoid layer. The $tanh$ layer ([-1,1]) creates new feature values which are used to update the potential new cell state, $\tilde{C}_t$. The output gate, o_t, 'filters' the cell state and determines how much of the updated cell state to pass through as an output. First the sigmoid layer decides which part to output, which is then passed through a $tanh$ function to map it between -1 and 1. C_t and h_t denote the activation vector for each cell and memory block, respectively.

[7] https://statistics.laerd.com/statistical-guides/spearmans-rank-order-correlation-statistical-guide.php

W_f, W_i, W_C, and W_o are the coefficient matrices. b_f, b_i, b_C, and b_o are the bias matrices. x_t are the model inputs.

2) 2D Convolutional LSTM: A *Convolutional Neural Network* (CNN) is a deep learning algorithm that can process images and 2D spatial data, learn to distinguish and extract features from the input data, and recognize patterns. The spatiotemporal map, as seen in Figure 1, is essentially a low resolution image consisting of a matrix of pixel values. The architecture of a CNN is based around sequential convolutional layers that apply relevant filters to the input data. A filter, also called kernel, is a 2D array of weights that detects certain patterns and is typically a 3x3 matrix. A convolutional operation consists of the kernels passing over the input image data and extracting features or patterns from the image. A single layer can typically detect low-level features such as edges or colors. With added layers, the network architecture is adapted to high-level features. The output is known as a feature map[8].

The LSTM layer can capture temporal patterns, a convolutional layer can capture spatial patterns. The ConvLSTM layers can process sequential and spatial information. The ConvLSTM has convolutional structures in the input-to-state and state-to-state (recurrent) transitions and applies convolutional operation instead of matrix multiplication [29]. By stacking several layers, a specialized network model is build for general spatiotemporal sequence forecasting problems. The formulations in Equation (3) illustrate the difference with the LSTM module as seen in Equation (2).

$$\begin{cases} f_t = \sigma\left(W_{xf} * \mathcal{X}_t + W_{hf} * \mathcal{H}_{t-1} + W_{cf} \odot \mathcal{C}_{t-1} + b_f\right) \\ i_t = \sigma\left(W_{xi} * \mathcal{X}_t + W_{hi} * \mathcal{H}_{t-1} + W_{ci} \odot \mathcal{C}_{t-1} + b_i\right) \\ \tilde{C}_t = \tanh\left(W_{xc} * \mathcal{X}_t + W_{hc} * \mathcal{H}_{t-1} + b_c\right) \\ \mathcal{C}_t = f_t \odot \mathcal{C}_{t-1} + i_t \odot \tilde{C}_t \\ o_t = \sigma\left(W_{xo} * \mathcal{X}_t + W_{ho} * \mathcal{H}_{t-1} + W_{co} \odot \mathcal{C}_t + b_o\right) \\ \mathcal{H}_t = o_t \odot \tanh\left(\mathcal{C}_t\right) \end{cases} \tag{3}$$

Where $*$ denotes convolutional operation and $\odot$ denotes an element-wise product. This formulation of the ConvLSTM by Shi et al.[29] is an extension to the LSTM in Equation (2), introduced by Gers [34]. The effect of this ”peephole” variant is the added cell state feedback at each gate ($W_c \odot \mathcal{C}_{t-1}$). The main difference between the ConvLSTM and the LSTM is that the inputs $\mathcal{X}_t$, cell states $\mathcal{C}_t$, hidden states $\mathcal{H}_t$, and gates i_t, f_t, and o_t are 3D tensors, of which the last two dimensions are spatial dimensions. Consequently, the multiplication between the weight matrices and input and the weight matrices and previous hidden states are replaced with convolutional operation ($W_x * \mathcal{X}_t$ and $W_h * \mathcal{H}_{t-1}$).

As merging with the vector output from the LSTM layer is required, the output of the final ConvLSTM is flattened to match the network model dimensions. A visual representation of a convolutional operation is shown in Figure 3, the exact model architecture is given in Section III-D4.

3) Data Preparation: The flight points and information input data represent information which is generally available to ATC:

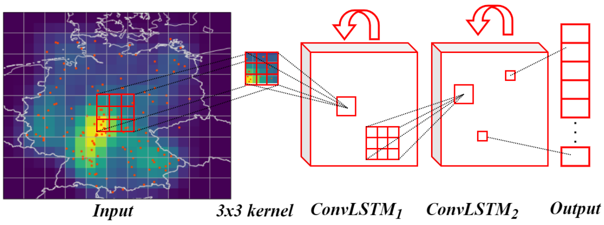


Fig. 3: General schematic structure of a convolutional operations of the ConvLSTM network branch.

1) **Actual flight points** Longitude, latitude, and FL at constant time intervals, at least 5 minutes prior to the first time instance of the prediction.
2) **Filed flight points** Longitude, latitude, and FL at constant time intervals. The total time spans from the time of the first actual flight point to the time at the longest desired look-ahead time. e.g. 5 minutes of actual flight points input and a 30 minutes look-ahead time require 35 minutes of filed flight points.
3) **Time stamps** Time of each flight point is formatted in hours as a float with 2 decimal points at each instance. The time stamp is only included as a variable for the actual flight points.
4) **Weekday** The day of the week upon arrival in the sector is formatted with hot-one encoding as a categorical variable.
5) **ICAO Flight Type** Scheduled or non-scheduled commercial operation is formatted with hot-one encoding as a categorical variable.
6) **STATFOR Market Segment** Various market segments of the flights are formatted with hot-one encoding as a categorical variable.

The spatial coordinates are transformed from the spherical coordinate system into 3D Cartesian coordinates, defined with respect to the local tangent reference frame with East, North, and Up (ENU) conventions and all numerical variables are Min-Max normalized. Bucketing is applied to the filed flight points because the filed flight points consists of a longer time frame. Next, the data is formatted according to the the LSTM input layer shape: [samples, time steps, features]. The length of the samples vector equals the number of flights in the data set, the time steps vector equals the number of actual flight points, and features vector equals the number of variables. Note that the length of the features vectors is larger than the six mentioned variables in the list due to bucketting of the filed flight points; Hot-one encoding of the categorical variables; and depends on which weekdays are part of the data set.

The air traffic dynamics feature maps are the input to the ConvLSTM layers and thus transformed into 5D tensor with shape: [samples, time steps, channels, rows, columns]. The length of the samples vector equals the number of flights. The time steps span the length of the filed flight points, but can have different time intervals. Therefore, the length of the vector is not necessarily identical to the length of the filed flight points vector. The channels refer to the number of features, which will be fixed to one at a time. The rows

[8] https://cs231n.github.io/convolutional-networks/#overview

and columns equal to the length of grid rows and columns as shown in Figure 1 and depend on the chosen grid size.

The output vector has the shape: [samples, time steps, features]. For the output, this means samples length equals the number of flights in the particular data set (training or testing), the time steps equal the desired look-ahead time, and the number of features is three (x, y, z).

Two methods for splitting the data set into training and testing sets are tested. The first method splits the data set based on whole days. A caveat, however, is that there could be a strong difference in flight patterns between days due to passenger behavior differences in, for example, weekends or weekdays. These patterns might not be properly recognized if there are no repeated days in the training and testing set. Therefore, the second method of splitting the data set is a randomized split, with a 75/25 ratio in size of training and testing data, respectively. A randomization of the flights might result in the model training with a unrealistic (lower) airspace occupancy.

4) Composite ConvLSTM Model Design: The deep learning network processes two input data sources and utilizes both LSMT layers as ConvLSTM layers, as introduced in Section III-D1 and Section III-D2, respectively. The composite ConvLSTM model architecture includes a variety of additional layers and processing steps for the model to perform as intended, as shown in Figure 4. The model is implemented using the Keras[9] library in Python, which runs on top of Tensorflow[10], an open-source platform for machine learning.

This model can be classified as a Multiple Parallel Input and Multi-Step Output predictor[11]. A model specifically developed for forecasting variable length output sequences (sequence-to-sequence) is called the Encoder-Decoder LSTM. The encoder is a model responsible for reading and interpreting the input sequence. Both *ConvLSTM2D* and the first *LSTM* layer constitute encoding models. All ConvLSTM layers are followed by a *BatchNormalization* layer in order to standardize the inputs to the next layer, reducing the sensitivity to initial weights and stabilizing as well as speeding up the learning process. Tests have shown that a second ConvLSTM layer leads to significantly improved prediction capabilities, but at higher computational costs. Nonetheless, the performance gain outweighs the computational costs. A *RepeatVector* is implemented to repeat the fixed-length output of the encoder so that the decoder part 'fits' to the encoder. The model must output a prediction for each time step in the output sequence, rather than a single prediction at the end of the sequence. Therefore, the decoding layers have the *return_sequence* set to *True*. *Dropout regularization* is implemented to reduce overfitting. *Flattening* allows the output shapes of the two branches to be matched prior to the *Concatenate* layer which merges the two branches. Finally, the *TimeDistributed* wrapper on the output layer is used to wrap a fully connected Dense layer. The *TimeDistributed* wrapper allows the same output layer to be reused for each element in the output sequence.

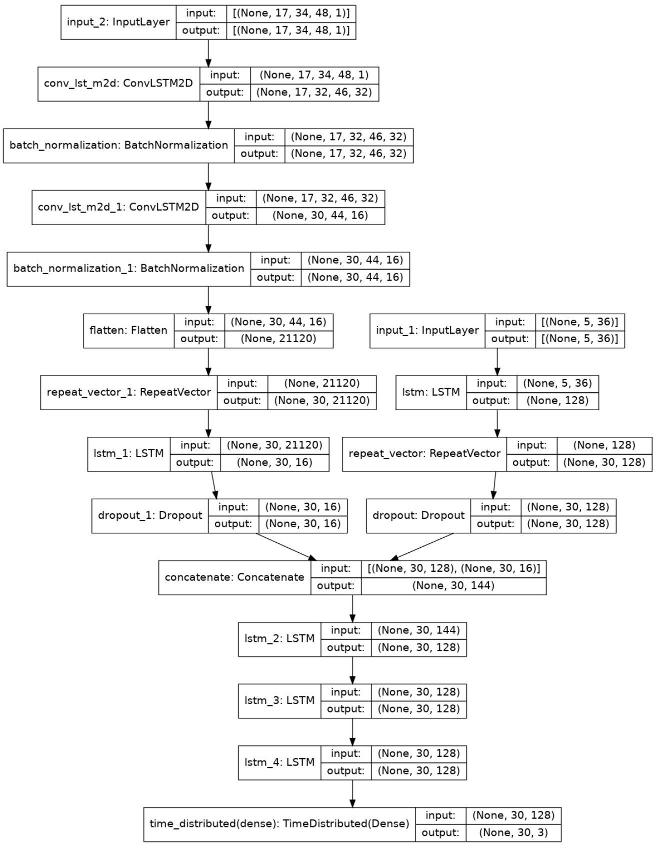

input_2: InputLayer | input: [(None, 17, 34, 48, 1)] | output: [(None, 17, 34, 48, 1)]
conv_lst_m2d: ConvLSTM2D | input: (None, 17, 34, 48, 1) | output: (None, 17, 32, 46, 32)
batch_normalization: BatchNormalization | input: (None, 17, 32, 46, 32) | output: (None, 17, 32, 46, 32)
conv_lst_m2d_1: ConvLSTM2D | input: (None, 17, 32, 46, 32) | output: (None, 30, 44, 16)
batch_normalization_1: BatchNormalization | input: (None, 30, 44, 16) | output: (None, 30, 44, 16)
flatten: Flatten | input: (None, 30, 44, 16) | output: (None, 21120)
input_1: InputLayer | input: [(None, 5, 36)] | output: [(None, 5, 36)]
repeat_vector_1: RepeatVector | input: (None, 21120) | output: (None, 30, 21120)
lstm: LSTM | input: (None, 5, 36) | output: (None, 128)
lstm_1: LSTM | input: (None, 30, 21120) | output: (None, 30, 16)
repeat_vector: RepeatVector | input: (None, 128) | output: (None, 30, 128)
dropout_1: Dropout | input: (None, 30, 16) | output: (None, 30, 16)
dropout: Dropout | input: (None, 30, 128) | output: (None, 30, 128)
concatenate: Concatenate | input: [(None, 30, 128), (None, 30, 16)] | output: (None, 30, 144)
lstm_2: LSTM | input: (None, 30, 144) | output: (None, 30, 128)
lstm_3: LSTM | input: (None, 30, 128) | output: (None, 30, 128)
lstm_4: LSTM | input: (None, 30, 128) | output: (None, 30, 128)
time_distributed(dense): TimeDistributed(Dense) | input: (None, 30, 128) | output: (None, 30, 3)

Fig. 4: Schematic representation of the architecture of the composite ConvLSTM model, a direct output of Keras.

In deep learning, anticipating the effect of each specialized layer and the hyperparameters can be a challenge. The (hyper-)parameters, tabulated in Table I, are selected by careful consideration of the model and data types as well as by simple sensitivity studies that observe the effect of experimentally changing the values.

TABLE I: Table with relevant (hyper-)parameters.

Parameter	**Layers**	**Argument**
Units	lstm_1	16
	All other	128
Dropout	All	0.25
Filters	conv_lst_m2d	32
	conv_lst_m2d_1	16
Kernel size	All ConvLSTM	3x3
Activation Function	All LSTM and ConvLSTM	*tanh*
Recurrent Activation	LSTM	*sigmoid*
	ConvLSTM	*hard sigmoid*
Optimizer		ADAM
Loss Function		MSE
Batch Size		64
Shuffle		False
Epochs		50

A batch size of 1 allows online learning. This means that after each training step, the network weights are updated and the true output of the previous time step is used to make a new prediction. In practice, this is computationally much too

[9] https://keras.io/api/layers/recurrent_layers/lstm/

[10] https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM

[11] https://machinelearningmastery.com/how-to-develop-lstm-models-for-time-series-forecasting/

heavy. Testing various batch has shown that a batch size of 64 only slightly deteriorates performance but allows a much better computational time per epoch. The only consequence is that it adds some instability to the learning process.

IV. Experimental Set-up of the Trajectory Prediction Phase

This section will specify the steps taken to test and evaluate the deep learning models, as proposed in Section III. First, the metrics used to present and analyze the results of the trajectory predictor are discussed. Next, the direct comparison between the air traffic dynamics and the three verification tests are explained. Fourth, some iterative model improvements are elaborated upon. Table II presents an overview of all the tests that are conducted for the trajectory prediction.

A. Model Evaluation

To assess the learning ability and behavior of the models; evaluate the prediction performance; and present the results, a variety of metrics and methods are applied.

1) Model Loss and Accuracy: The loss and accuracy functions of the training and validation data sets are evaluated after each run. This plot is used to evaluate the training progress, convergence speeds as well as stabilization, and over- or underfitting of the model. The loss function corresponds to the Mean Square Error and measures the error in the learning process. The model accuracy calculates the percentage of predicted values that match the actual values.

2) Prediction Accuracy: Three conventional navigation-error metrics are used evaluate the model output. The flight level error is a direct comparison between the actual flight points and predicted model output. The prediction accuracy in the horizontal plane is evaluated by measuring the *Cross-Track Error* (CTE) and the *Along-Track Error* (ATE). The horizontal errors are visualized in Figure 5. These are presented as a box-plot for each look-ahead time. The box-plots include the median as well as the outer bounds of the prediction accuracy.

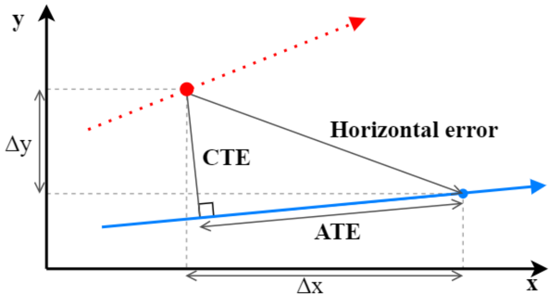


Fig. 5: The horizontal performance metrics between the actual flight path (red) and predicted flight path (blue).

3) Visual Inspection: A random selection of flights is translated to a map to visually inspect the difference between input flight points, input filed flight points, the predicted flight points, and the actual flight points.

4) Computational Effort: The models are trained on a GPU (NVIDIA K80) backed server provided by Kaggle. This significantly speeds up the training process but has time and RAM limits on allowed usage. The time taken per epoch is used to compare training speeds per model, which helps assess the model efficiency.

B. Direct Comparison between Air Traffic Dynamics Features

The composite ConvLSTM network is used to compared the performance of the selected air traffic dynamics features. The purpose of these tests is to observe the difference in trajectory prediction accuracy for the various air traffic dynamics features. The tests include feature grids with randomized values and all-zero values to test if the air traffic dynamics features have any effect on the prediction accuracy. The comparison is performed with identical models and identical random seeds.

As mentioned in Section III-D, the model is tested with two types of data set splits: based on whole days and randomly sampled flights. However, certain distributions of the whole day data split lead to significant overfitting. An example model suffering from overfitting can be found in Appendix A. The prediction accuracies of these runs cannot be reliably compared with the runs that do not suffer from overfitting. Therefore, experiment 1a -as seen in Table II- includes only days 2 and 3 as training days and day 1 as the testing day and does not suffer from overfitting.

In order to further mitigate this effect as well as to verify the difference between air traffic dynamics features, the experiment is repeated with some changes to the experimental set-up. Test 1b includes seven days of flight data; a random split of training and testing flights at a 0.75/0.25 ratio; and 2 minute time intervals between points as well as feature grids. This experiment corresponds to test 1b in Table II. Note that test 1a does not include the results of the randomly sampled data split due to a similarity in results to test 1b.

C. Baseline Experiment

A baseline model is used to further test the effects of the air traffic dynamics features on the accuracy of the trajectory prediction. The baseline model is a fully-connected LSTM network that is essentially a stack of the right hand branch of the composite ConvLSTM model with the three remaining LSTM layers, as shown in Figure 4. The baseline model can be classified as a stacked LSTM network and has the same *Dropout* layer as well as a *TimeDistributed* wrapper to wrap a fully connected *Dense* layer. The input is identical to the flight point and information input data as used in the main experimental model 1b, using a randomly sampled data split only. The (hyper-)parameters are equal to those relevant to the LSTM layers as presented in Table I. The set-up of the baseline experiment is presented as test 2a in Table II.

D. Verification

1) Accuracy of Filed Flight Points: The accuracy of the filed flight plan is considered to evaluate the deep learning model effectiveness. A deep learning model -with the filed flight points as inputs- can be expected to predict the flight points with at least the same accuracy as the raw filed flight points. This test correspond to test 3 in Table II.

2) Prediction Model Simplification: A simplified composite ConvLSTM model is also tested. The idea is that this will reduce noise and model error and thus make differences between air traffic dynamics features more distinguishable. The aim of this verification test is to test if the air traffic dynamics features contain information that can be recognized by the ConvLSTM layers. This model does not include any flight information nor filed flight points but does contain 5 minutes of actual flight points (x,y,z) prior to the prediction. The prediction only consists of a single predicted flight point at 24 minutes look-ahead time. The runs are performed with the selected air traffic dynamics features as well as all-zero grid values. See test 4 in Table II for the details.

E. Iterative Model Improvements

During the tuning, testing, and result collection phase, observations and analyses are continuously made that lead to model improvements. Nevertheless, not all intermediate results are discarded as some direct comparisons still reveal certain insights into the effect of air traffic dynamics and on the model performance. For that reason, Section VI includes intermediate results that do not necessarily correspond to the most optimal model design or set-up, as seen with test 1a. Moreover, this section presents an improvement made to the baseline model.

1) Initialization Error: The models of test 1a, 1b, and 2a used to comparing the air traffic dynamics features suffer from unexpected initialization errors. The observation is classified as a model error because the predictions at short look-ahead times should correspond closely with the actual flight points that are part of the model input data. A simple linear extrapolation of the input flight points would produce better initial predictions than most of the observed predictions with initialization error. These errors are highlighted in Section VI and are visualized for further reference in Appendix B.

2) Improved Baseline Model: To better explain the results of the deep learning models as well as to mitigate the initialization error, an improved version of the baseline model is included in the results: test 2b in Table II. To make a reliable comparison between the baseline model and the ConvLSTM model, the improved baseline model is not used in the assessment of the air traffic dynamics features. The following two improvements are made to the baseline model.

First, the length of the input time vector is increased. The initial input time vector includes 6 minutes of data, corresponding to a sequence of 3 data points. The input sequence is too short for the model to accumulate enough information over the time inputs. To mitigate this limitation, the length of the input sequence is increased to 6 points, corresponding to 12 minutes. Consequentially, the total considered time frame of each flight is increased from 35 to 42 minutes, which significantly reduces the number of selected flights from the data set[12]. To mitigate the loss of data, the data set is extended to include 2 weeks of flights and the look-ahead time is reduced to 24 minutes. Second, the default bias initializer generates tensors intitialized to zero. This is sufficient when long time sequence inputs allow a faulty initialization to be corrected by accumulating enough information. However, in the case of short input sequences, this is not the case. To make the model more responsive to initial weights and speed up learning, the bias tensors are initialized to one.

TABLE II: An overview of all the tests that are conducted in Section VI, including variations to the model.

Test	Model/Test Type	Data set [days]	Train/Test Split	Interval [minute]	Input Sequence Length	Output Sequence Length
1a	ConvLSTM	3	Whole days	1	5 points	30 points
1b	ConvLSTM	7	Rand 0.75/0.25	2	3 points	14 points
2a	Baseline	7	Rand 0.75/0.25	2	3 points	14 points
2b	Improved Baseline	14	Rand 0.75/0.25	2	6 points	12 points
3	Filed Flight Point CTE	7	-	2	-	17 points
4	Simplified	7	Rand 0.75/0.25	1	5 points	1 point at 24 mins

V. Results of the Statistical Analysis

This section presents the results of the statistical analysis by first analyzing the rank correlations followed by a cross-correlation analysis. In Section V-C an evaluation of the statistical analysis is performed.

A. Rank Correlation

The results of the Spearman test and the corresponding p-value with a grid size of 0.5 degrees on a geodetic coordinate system are given in Table III and Table IV. The bandwidth of the KDE is set to three times the grid size.

TABLE III: Spearman's coefficient of rank correlation for the instantaneous features at 0.5 degree grid size

Feature	Track Deviation		FL deviation	
	ρ	p-value	ρ	p-value
TD	0.540	0.000	0.005	0.130
ACC	0.341	0.000	-0.014	0.062
HdgCnt	0.060	0.000	0.030	0.000
HdgCntKDE	-0.178	0.000	0.008	0.129
HdgAvg	0.118	0.000	0.050	0.000
SpdCnt	0.063	0.000	0.013	0.072
SpdCntKDE	0.297	0.000	0.007	0.097
SpdAvg	0.048	0.000	0.034	0.000
AltCnt	0.063	0.000	-0.029	0.000
AltCntKDE	0.255	0.000	-0.007	0.106
AltAvg	0.048	0.000	-0.019	0.009
MinDst5	0.140	0.000	-0.008	0.286
MinDst5KDE	-0.215	0.000	-0.008	0.137
MinDst10	0.175	0.000	-0.016	0.024
MinDst10KDE	0.067	0.000	-0.002	0.658
MinDst50	0.200	0.000	-0.011	0.145
MinDst50KDE	0.475	0.000	0.004	0.238
AvgDst	-0.080	0.000	0.001	0.864
SDHdg	0.286	0.000	-0.009	0.213

[12] The data selection from the raw data source includes several filters in order to select suitable en-route flights. For example, the flights must remain above a certain FL to be inside the UAC.

TABLE IV: Spearman's coefficient of rank correlation for the aggregated features at 0.5 degree grid size

Feature	**Track Deviation**		**FL Deviation**		**Delay**	
	ρ	**p-value**	ρ	**p-value**	ρ	**p-value**
TD	0.433	0.000	-0.003	0.831	-0.048	0.000
ACC	0.464	0.000	-0.004	0.760	-0.044	0.001
HdgCnt	0.222	0.000	0.043	0.002	-0.031	0.022
HdgAvg	0.376	0.000	0.057	0.000	-0.031	0.023
SpdCnt	0.303	0.000	0.014	0.294	-0.061	0.000
SpdAvg	0.363	0.000	0.041	0.002	-0.031	0.024
AltCnt	0.174	0.000	-0.032	0.017	-0.114	0.000
AltAvg	0.142	0.000	-0.005	0.734	-0.114	0.000
MinDst5	0.182	0.000	-0.024	0.080	-0.049	0.000
MinDst10	0.253	0.000	-0.018	0.197	-0.040	0.004
MinDst50	0.361	0.000	-0.009	0.492	-0.056	0.000
AvgDst	0.545	0.000	0.021	0.129	0.010	0.453
SDHdg	0.320	0.000	0.001	0.942	-0.037	0.006
Size	0.547	0.000	0.020	0.142	-0.016	0.238

B. Cross-Correlation

Cross-correlation testing is performed between the dependent variables calculated for each grid at each instant as well as aggregated over each flight. The analysis points out that from the instantaneous measures of the variables, the following variables are correlated, of which the latter is discarded: *AltCnt* and *AltAvg*; *TD* and *ACC*; *AvgDst* and *ACC*; *MinDst50* and 'KDE of *MinDst50*'. The analysis of the aggregated measures of the variables shows correlation between *AvgDst* and *Size*, of which *Size* is discarded as a variable. *MinDst50*, *ACC*, and *TD* are all highly correlated, *MinDst50* and *ACC* are discarded as variables.

Last, based on the cross-correlation analysis it is argued that the KDE of the variables cannot be reliably evaluated. Some grid values that are zero for non-KDE features contain a value for KDE features due to the KDE-distribution. Consequently, the KDE-features have increased rank correlation at these grids that were previously zero. As a result, the KDE of the variables are cross-correlated. However, these correlations do not form a causal relationship but are a statistical flaw that introduces bias. Therefore, the KDE correlations are excluded from the evaluation of the feature rank correlation.

C. Evaluation of Statistical Analysis

First, the following caveats to the statistical analysis must not be overlooked. The independent variables of the statistical analysis are not identical to the output variables of the predictive model, thus a discrepancy is expected to remain. Correlations do not imply causality: this is observed with the cross-correlated KDE of the variables and also between the highly correlated *Size* and *AvgDist* variables for aggregated features. Moreover, bivariate testing is not indicative of possible trivariate (or higher) relationships between the metrics and features. It is expected however, that multivariate relationships can be captured by the composite ConvLSTM network.

None of the computed features have a convincing statistical monotonic relation with FL deviation. 14 of the 19 variables for individual grid features do not pass the null-hypothesis. The aggregated variables lead to similar outcomes. This suggests that the FL deviations are insignificant and/or randomly distributed. ATCOs and pilots are not inclined to resort to vertical manoeuvres to de-conflict air traffic situations and at the same time, the deviations are very small. The vertical separation requirement is much smaller in magnitude compared to the horizontal, so any vertical manoeuvre is small. As a consequence, predicting the vertical component of the trajectory is not a priority and is kept separate from the horizontal component during performance evaluation.

The results of the delay metric suggest that there are no significant correlations between the air traffic features and delay. A majority of the variables pass the null-hypothesis, but the magnitude of the correlations are mostly insignificant. Multiple variations to the delay metric have been tested with similar outcomes. Two explanations can be attributed to this unexpected result, which are not mutually exclusive. First, computing delay depends on the irregular sector boundaries and all flights have different entry and exit points. This is even the case between the actual and filed flight points for the same flight. These statistical impurities may cause an excess of noise. Second, the results could mean that for en-route traffic, delay is not significantly affected by the surrounding air traffic dynamics. This conclusion does not coincide with the expectation. Therefore, this may be interpreted as an indication that the relation between the air traffic features and temporal effects are highly non-linear and/or multivariate.

The results of the horizontal track deviation support the research question and lead to valuable insights that are applied during the trajectory prediction. The individual features per grid show that the standard deviation of the heading angle and traffic density have the highest rank correlation. Generalizing, the remaining KDE of the features have higher correlation, but are disregarded as mentioned before. Especially with the aggregated features, significant correlations can be observed with the track deviation. The difference between individual grid feature values and aggregated values verify that there is a strong time-varying effect between feature grid value and track deviation. This verifies the suitability of the LSTM layers which capture temporal effects.

Comparing the results of different grid sizes suggests that the grid size of 1.5°has consistently higher rank correlations with track deviation, for both the individual as well as the aggregated features. However, an increased number of flights in a grid also translates to increasing the value of many complexity feature values. This can be confirmed by the increased cross-correlation at large grid sizes. The grid size is an important parameter because it can capture the consideration horizon of a pilot or ATCO. In other words, the grid size parameter addresses the question: What distance ahead of the flight direction is considered when making tactical route decisions? The use of convolutional layers is especially suitable for mitigating the risk of choosing the 'wrong' grid size as any feature/pattern is extracted, regardless the grid size. Therefore, the smallest grid size is chosen: 0.5°.

The following air traffic dynamics features have a statistically significant correlation to the flown trajectory. These features are most likely to include information which can be extracted by the ConvLSTM network, in order of highest to lowest likelihood:

- **Traffic Density** (*TD*)
- **Heading Variance** (*SDHdg*)
- **Mean Separation** (*AvgDst*)
- **Average Speed Change** (*SpdAvg*)
- **Average Heading Change** (*HdgAvg*)
- **Altitude Change Count** (*AltCnt*)

TD is primarily used to test, tune, and verify the composite ConvLSTM network as it has the most convincing correlation. *SDHdg* and *AvgDst* are also included in the trajectory predictions. The three remaining features have relatively weak correlations, it is not expected that these will result in a increased trajectory prediction accuracy and are therefore not included henceforth.

VI. Results of Trajectory Prediction

This section presents the results as well as brief analyses of the composite ConvLSTM model and the various verification tests. The results are presented in correspondence with Table II.

A. Composite ConvLSTM Model Trajectory Prediction

1) Test 1a: Three Days Data Set: The loss function scores are summarized in Table V. The loss scores with *TD* and *SDHdg* are lower compared to the random and all-zero grid values.

TABLE V: Loss score with three days of data comparing the air traffic features. Test 1a in Table II.

Run	Feature	Data split [days]		Loss score	Overfit
		Training	Testing		
1	*TD*	2,3	1	0.00032	-
2	*SDHdg*	2,3	1	0.00053	-
3	*AvgDst*	2,3	1	0.00140	-
4	Random	2,3	1	0.00077	-
5	All zeros	2,3	1	0.00070	-

For the example shown in Figure 6, the loss function reveals that the model does not suffer from overfitting, albeit strong fluctuations are visible in both the loss and accuracy function. These fluctuations are expected to cause the difference in loss scores between the different runs seen in Table V.

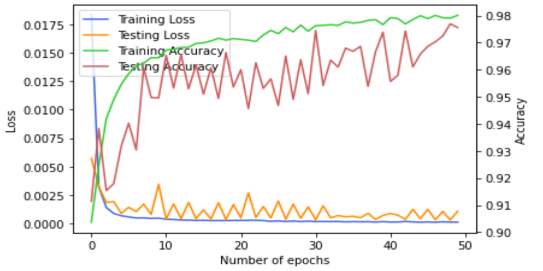


Fig. 6: The loss and accuracy corresponding to run 1 in Table V.

Figure 7 presents the prediction accuracy for the models with *TD*, random, and all-zero feature grid values. 75% of the predictions with *TD* feature values have a CTE under 30 NM at 30 minutes look-ahead time, with a median slightly above 20 NM. At 30 minutes look-ahead time, the prediction accuracy is similar for the remaining features, except the *SDHdg* which has slightly better performance (shown in Appendix C). The distribution of the CTEs versus look-ahead time is different for all models, except with *TD* and the random grid values. A noteworthy observation in Figure 7c is that the all-zero grid features produce much smaller CTEs at lower look-ahead times. For this run, the initialization error is clearly observed. For the remaining features, the variation in performance is expected to be due to model randomness, as seen in Figure 6. The computational time per epoch is around 80 seconds.

TABLE VI: Loss score of the composite ConvLSTM model with seven days of data. The training and testing data set includes whole-day splits as well as random splits at a 0.75/0.25 ratio to study the effect that full days have on the prediction accuracy. Test 1b in Table II.

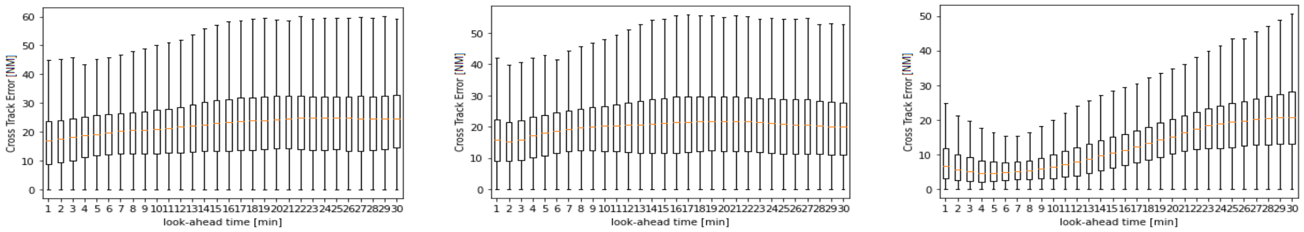
Run	Feature	Data split [days]		Loss score	Overfit
		Training	Testing		
1	*TD*	1,2,3,5,7	6,4	0.00077	-
2	*TD*	1,2,3,4,7	5,6	0.00180	Yes
3	*TD*	2,3,4,5,7	1,6	0.00074	-
4	*TD*	2,4,5,6,7	1,3	0.00079	-
5	*TD*	1,2,4,5,6	3,7	0.0019	Yes
6	*TD*	1,2,3,4,5	6,7	0.00539	Yes
7	*TD*	rand 0.75	rand 0.25	0.00009	-
8	*SDHdg*	rand 0.75	rand 0.25	0.00011	-
9	*AvgDst*	rand 0.75	rand 0.25	0.00086	-
10	*random*	rand 0.75	rand 0.25	0.00105	-
11	*zeros*	rand 0.75	rand 0.25	0.00013	-

2) Test 1b: Seven Days Data Set: Comparing Figure 6 and Figure 8, the model loss and accuracy functions are much improved and more consistent when using seven days of data and randomly sampled sets. Moreover, it is confirmed that whole day data set splits lead to worse training behavior and a higher tendency to overfit, as seen in runs 1-6 compared to runs 7-11 in Table VI. This is exaggerated when the two testing days are consecutive days.

Figure 9 presents the prediction accuracy of the models with *TD*, random, and all-zero grid values. The *TD* feature (Figure 9a) leads to the best prediction accuracy at long look-ahead times: 75% of the predictions have CTEs well under 15 NM at 28 minutes look-ahead time, with a median around 8 NM. However, the model with the all-zero feature grid values can be seen to produce more accurate predictions at short look-ahead times ($<$ 10 mins). The results of the *SDHdg* and *AvgDst* are shown in Appendix C. The *AvgDst*, *SDHdg*, and random grid values all generate predictions that are very similar to one another, but with worse accuracy compared to *TD* and the all-zero grid values. The computational time per epoch is around 130 seconds.

In both Figure 9a and Figure 9b, the initialization error is observed. The highest prediction accuracy is achieved between 6 and 10 minutes of look-ahead time. This observation is also made for the *SDHdg* and *AvgDst* feature models.

3) Analysis of Composite ConvLSTM Model: The overall prediction accuracy of the model with the seven day data set improved the prediction accuracy significantly, seen by



(a) CTE of the model with *Traffic Density* grid values. Run 1 in Table V.

(b) CTE of the model with random grid values. Run 4 in Table V.

(c) CTE of the model with all-zero grid values. Run 5 in Table V.

Fig. 7: A difference in prediction accuracy can be observed between the models incorporating random and all-zero grid values.

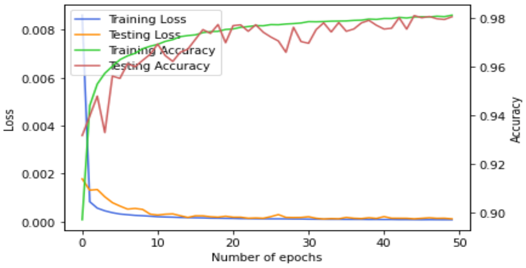


Fig. 8: The loss and accuracy corresponding to run 7 in Table VI.

comparing the corresponding sub-figures from Figure 7 with Figure 9. Increasing the number of days beyond one week however, exceeds the computational limits.

It is confirmed that using non-consecutive testing days lead to better results (Run 1 versus 6 in Table VI). The significant difference in training ability as well as prediction accuracy between certain variations of data set splits based on whole days suggests that there are temporal effects between days that must not be overlooked. The temporal effect between days is likely caused by different flight traffic patterns (e.g. weekend- versus week-days).

Furthermore, it is confirmed that randomly sampling the flights greatly improves the model learning behavior as well as the prediction accuracy. In this approach, reoccurring flight patterns on each day are taken into consideration. In the real world, this translates to making predictions based on reference observations that occur simultaneously or in the future, which is not realistic. Nevertheless, this approach makes the results much more consistent and makes differences between air traffic dynamics features more apparent. Thus, seven days of data with randomly sampled testing and training sets allow for a more reliable comparison between the air traffic dynamics.

The analysis of the performance between the air traffic dynamics features is based on the seven day data set with randomly sampled sets. The difference between *AvgDst*, *SDHdg*, and random grid values is not observable, which indicates that the *AvgDst* and *SDHdg* do not contribute to the trajectory prediction. The *TD* performs marginally better compared to the all-zero grid value model at higher look-ahead times. Based only on these results, the inclusion of traffic density appears to improve the trajectory prediction accuracy compared to the other features as well as without any air traffic dynamics information. It is noteworthy that the all-zero grid values model outperforms the other feature grid values as well as the random grid values. Thus, it is possible that non-zero grid values make it 'more difficult' for the model to extract information from the remaining input data. This would indicate a limitation to the model and not necessarily mean that the remaining air traffic features do not contain useful information.

Last, a brief remark on the initialization error. As mentioned in Section IV-E, the initialization errors are counter-intuitive because the initial predictions are expected to be more accurate than the predictions at longer look-ahead times. Therefore, the error indicates that there is some degree of model error or the model is unable to extract the useful information from the input data. The absence of initialization error in Figure 9c is not observed for all repeated runs, which are not all shown here. These observations indicate that the initialization error is inherent to the model and not to the air traffic dynamics features.

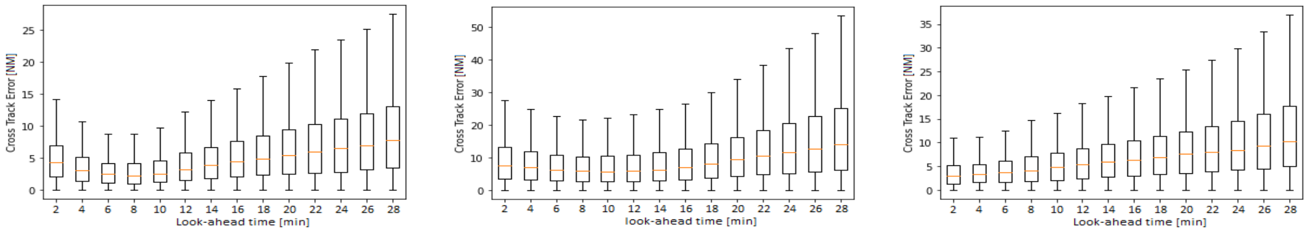
Further analysis in the following sections is needed to understand if the difference in performance is due to (random) model error or if there is indeed a measurable improvement in trajectory prediction accuracy from the inclusion of *TD*.

B. Baseline Experiment

1) Test 2a: Baseline model: The results of the baseline model are shown in Figure 10. 75% of the predictions have CTEs slightly under 20 NM at 28 minutes look-ahead time, with a median around 10 NM. The computational time per epoch is 4 seconds.

Compared to Figure 9a, the prediction accuracy of the composite ConvLSTM with the *TD* feature is better beyond 10 minutes of look-ahead times. However, the results depicted are an average representation of the repeated experiments, which cannot all be visualized in this paper. Two examples of better and worse performance of the baseline model are shown in Appendix D. Therefore, it cannot be reliably concluded that there is a observable difference in trajectory prediction accuracy between the ConvLSTM model with *TD* and the baseline model, test 2a. The baseline model also suffers from the initialization error.

2) Test 2b: Improved Baseline model: Figure 12 depicts the results of the improved baseline model, including the loss function, CTE, ATE, and FL deviation for completeness. The initialization error is reduced and the CTE is more constant



(a) CTE of the model with *Traffic Density* grid values. Run 7 in Table VI.

(b) CTE of the model with random grid values. Run 10 in Table VI.

(c) CTE of the model with all-zero grid values. Run 11 in Table VI.

Fig. 9: CTE of trajectory prediction using seven days of data and randomly sampled data sets. Note that the sampling time is two minutes, resulting in a maximum look-ahead time of 28 minutes.

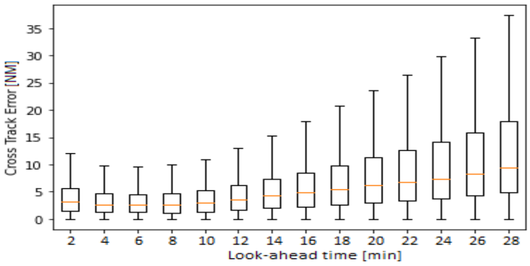


Fig. 10: The CTE of test 2a: The baseline model with identical conditions to test 1b as seen in Figure 9.

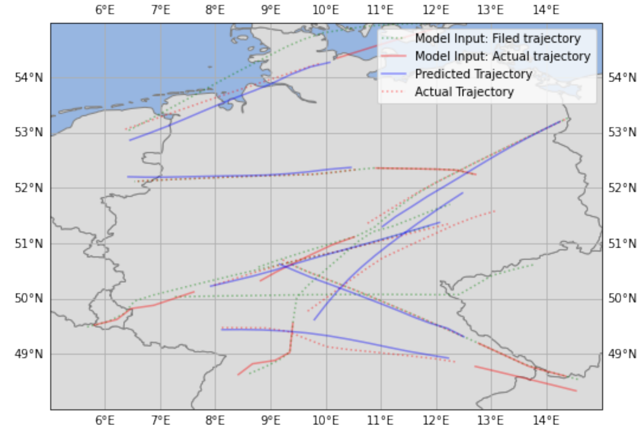


Fig. 11: A random selection of predicted trajectories corresponding to test 2b.

compared to test 2a. This is especially true for the upper quartiles and extremities of the CTEs. 75% of the predictions have CTEs below 10 NM at 24 minutes look-ahead time, with a median around 5 NM. The ATE follows a similar pattern compared to the CTE. The FL error is small: The FL error of 75 % of the predictions at 24 minutes of look-ahead time are below 400 meters. The loss function shows that the model learns well and does not suffer at all from overfitting. At 24 minutes look-ahead time, the performance is very similar to the composite ConvLSTM with the *TD* feature.

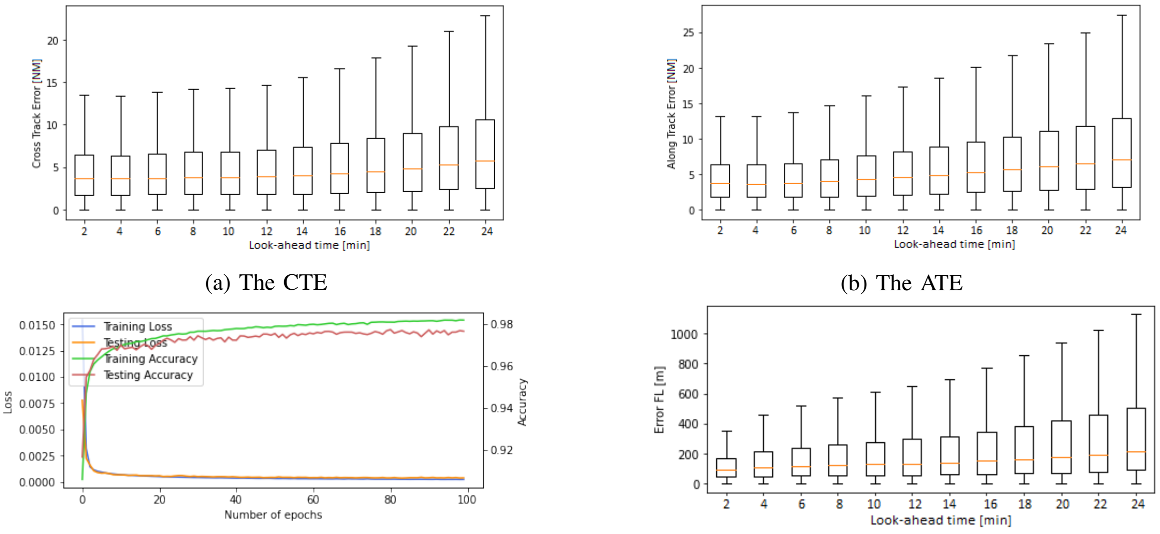
Figure 11 presents a random selection of trajectories, translated on a map. The initial prediction is generally accurate. Two observations stand out: First, for flights that deviate far from the filed trajectory, the predicted trajectories outperform the filed trajectories significantly (e.g. the flight starting furthest south-east, above Austria). Second, for flights that gradually approach the filed flight path, the predictions tends to follow the initial deviation of the filed flight paths. Thus, the prediction diverges over time (e.g. the northern most flight).

To conclude, it is suspected that the marginal differences between the baseline and composite ConvLSTM model are a consequence of the model and possibly GPU backend library randomness. The baseline only captures the difference within similar LSTM-based models -with or without air traffic dynamics-. However, the baseline comparison does not verify the model effectiveness to this problem. The highly complex air traffic dynamics input data and model randomness make it difficult to reliably assess the model effectiveness and the thus also the effect of air traffic dynamics on trajectory prediction. Therefore, the verification tests are needed.

C. *Verification*

To further analyze the model and air traffic dynamics, this section includes two verification tests to: (1) Compare the prediction accuracy directly to the accuracy of the filed flight points in order to verify the model learning capabilities. (2) Isolate the air traffic features and observe any effect between the prediction accuracy and the various air traffic features.

1) Test 3: Accuracy of the Filed Flight Points: The CTE and ATE of the filed flight points are shown in Figure 13. The median of CTE of the filed flight points slightly above 0 NM and thus much lower than any previous trajectory prediction. However, in comparison with the CTE of test 2b (Figure 12a) and test 1b with *TD* (Figure 9a), the upper quartile values and the extremities are consistently higher for the filed flight points. This reveals that the both the composite ConvLSTM and the stacked LSTM-network are able to make better trajectory predictions when the filed flight points are significantly deviated from the actual flight points. This coincides with the observations made in Figure 11. For many flights however, the filed flight path coincides with the actual flight path, as also seen in Figure 11. For these flights, the trajectory prediction cannot be more accurate than the filed flight path. A well-functioning prediction should match this accuracy for these types of flights. As for the ATE, the predictions of the improved baseline model are nearly identical to the ATE of the filed flight points.



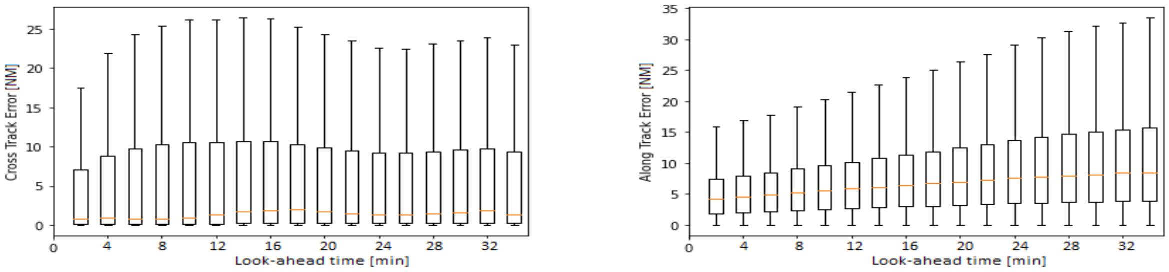
(a) The CTE

(b) The ATE

(c) Loss and accuracy of testing and training data

(d) The flight level error

Fig. 12: The trajectory prediction performance for the improved baseline model, corresponding to test 2b. For completion of results, all figures of performance metrics are included.



(a) Absolute value of the CTE of the filed flight points.

(b) Absolute value of the ATE of the filed flight points.

Fig. 13: A direct comparison between the filed flight points and actual flight points for seven days fight data.

2) Test 4: Simplified Model: The results of test 4 are summarized in Table VII[13].

TABLE VII: The RMSE with the seven days sets for four features grid values.

Run	**Feature**	**RMSE**		
		***x* [NM]**	***y* [NM]**	**Altitude [m]**
1	*TD*	42.6	40.1	2298
2	*SDHdg*	57.1	55.2	2388
3	*AvgDst*	57.3	57.3	2397
4	*zeros*	54.9	54.7	2375

The absolute prediction accuracy is poor compared to the previously tested models. However, this coincides with the expectation because the filed flight points are not included in this model. There is a measurable increase in prediction accuracy in the horizontal plane with the *TD* feature compared to the remaining features, including the all-zero grid values.

The purpose of the simplified model is to test if the ConvLSTM layers are reactive to spatiotemporal feature maps and use the information embedded in the grids to produce a prediction. This is confirmed by the results. Consequently, the conclusion of the statistical analysis can also be (partially) verified through the simplified model: there exists an observable relationship between the surrounding traffic density and the flown trajectory; in addition, the vertical component of an en-route flight is not affected by the surrounding air traffic dynamics. However, there is a possibility that the model learns to predict towards grids with higher *TD*, as a higher *TD* increases the likelihood that the flight in question passes through that area.

VII. Discussion

The discussion will emphasize two aspects of this study, in line with the research objective.

A. The Effect of Air Traffic Dynamics on Trajectory Prediction

Based on the observations made in this research, it is likely that the air traffic dynamics -especially traffic density- can

[13] Because the prediction and the reference actual flight point are single points instead of trajectories, the CTE and ATE cannot be calculated. Instead, the navigational accuracy metric is the RMSE in 3D Cartesian coordinates defined w.r.t. the local tangent reference frame (ENU).

be used to improve trajectory predictions. The following three observations are made to support this statement. First, the traffic density has a statistically significant correlation with track deviation. Second, the traffic density causes an occasional improvement in prediction accuracy in direct comparison to the baseline model. Third, the inclusion of traffic density improves the predictive capabilities of the simplified model. These three observations are the result of different tests. The fact that the results point in the same direction makes it likely that the traffic density can contribute to an increased accuracy of a trajectory predictor. However, the contribution is expected to be very small because the solution space of an en-route flight is very large at medium- to long-term look-ahead times. This means that the required adjustments in flight path or in speed for (medium- to long-term) conflict avoidance are in the order of a few degrees or knots.

To build upon the discussion above, the very small flight path or speed adjustments made by en-route flights in response to air traffic dynamics has two potential consequences. First, it is likely that both the statistical analysis and the deep learning model are not sensitive enough to detect such marginal adjustments and simultaneously correlate this to the air traffic dynamics. Second, it is difficult to detect the small effect on trajectory predictions consistently due to model variations and because the filed flight points dominate the trajectory prediction. A solution, which is discussed in greater detail in Section IX, is to apply increased filtering of flights.

The selected data set purposely includes highly congested days. It is possible that this actually made it more difficult to find a relation between the air traffic dynamics and the flight trajectory behavior because a ‘busy’ day means that pilots must adhere more strictly to the procedures and ATCO commands. It is possible that during ‘quiet’ days the effect is more significant because pilots have more freedom to fly their desired flight path. However, the chosen time-frame includes the night, with low levels of air traffic. Therefore, the chosen data set should reveal both patterns of flight behavior. Nevertheless, both the statistical analysis and the trajectory prediction might reveal different insights for scenarios exclusively with low amounts of air traffic.

The difference in model behavior depending on which days were part of the training and testing set indicate that the temporal effects between days are significant. It is possible to exploit these effects for trajectory prediction, but the data set must then include repeated days of the week. However, taking multiple weeks of data in order for each day to reoccur and allow for repetitive training was not possible due to computational limits. A different scheme of selecting, filtering, and interpolating data would be necessary in order for the computational effort to be within acceptable bounds. An alternative, for example, is to take flight data from all Mondays during two or more consecutive months.

It is a challenge to determine the causality of the correlation found between traffic density and predicted flight points (test 4). The deep learning model may learn to predict a trajectory to move towards a region of higher traffic density because there is a higher likelihood that any aircraft will pass through that region. This challenge will remain, even if a future model does improve the predictions by using air traffic dynamics.

Last, there is a difference in the results between the statistical analysis and the deep learning model. The statistical analysis reveals that there is a monotonic relation between the three selected air traffic dynamics features and the horizontal track deviation. The deep learning model is only able to verify a relationship with traffic density. This is in-part because different metrics were used to evaluate the trajectory. Nevertheless, it can be argued that the statistical analysis is of added value to this research. It would be a black-box if the features were not analysed prior to the trajectory prediction phase. Even though deep learning models are celebrated because heavy filtering and structuring of the input data are often not needed, the two-step method applied in this study allows a better understanding of the relationship between the air traffic dynamics and track deviation.

B. Suitability of LSTM-based Model to Predict 4D Flight Trajectories

The composite ConvLSTM model with the chosen input features does not observably improve the accuracy of the 4D trajectory prediction, based on the conditions of this research: beside the small flight track and speed adjustments, it is assumed that three limitations to the model contribute to this conclusion. First, the randomness and variability of the results during repeated experiments makes reliable observations on the performance of the models a challenge. Second, the difference between the composite ConvLSTM model performance with zero and non-zero grid values indicate that the model has difficulty to extract information from the spatiotemporal input data. Third, the extent to which the filed flight points dominate the trajectory prediction overshadow any contribution of the air traffic dynamics. Therefore, based on the evidence in this study, the contribution of air traffic dynamics into the composite ConvLSTM model does not outweigh the factor 40 increase in computational time per epoch.

The increase in performance of the improved baseline model means that the tested composite ConvLSTM model is suboptimal. It can therefore be questioned if the effect of air traffic dynamics is indeed not observable due to the above explained reasons or if the model does not yet perform adequately. Evidently, it can be recommended to improve the composite ConvLSTM model accordingly and repeat this research. However, this has not been done for several reasons. First, one of the two baseline model improvements involves increasing the input sequence length. As explained in Section IV-E, two or more weeks of flights have to be selected in order to maintain an acceptable look-ahead time as well as a sufficient amount of flights. A different and larger selection of flights means that the air traffic dynamics features have to be recomputed and the size of the input tensor grows significantly. Consequently, this exceeds the computational limits. There are solutions to mitigate this limitation: a larger time interval for data interpolation can be applied; a new scheme of data selection based on flight patterns or minimum track deviation can be designed; the grid size can be increased. The solutions would alter the input data set to an extent that comparing the results to the current test 1,2 and 3 is no longer reliable and all the experiments in this research must be repeated. Therefore, it is recommended to process these model improvements in a future, follow-up study.

Next, under the conditions of this research, both models are capable of generating better trajectory predictions compared to the filed flight path. The 'shape' of the flown path as well as the magnitude of deviation from the filed flight path have a significant effect on the performance of the predictor. For example, the deep learning models do improve the prediction accuracy for flights that have a very inaccurate filed flight path. Thus, multiple separate, specialized predictive models, based on certain flight path characteristics, could lead to better results: this is further discussed in Section IX. Moreover, at en-route flight speed, small errors in the initial prediction accumulate to large spatial errors with 30 minutes of look-ahead time. With live air traffic data, regular model updating would ensure that predictions under 30 minutes of look-ahead time do not ever diverge to an unacceptable extent. Similar to online learning, this can mean that the model is updated with the new actual flight points every couple minutes or when the prediction error exceeds a threshold.

An uncertainty which potentially inhibits the model from learning are the time-vector discrepancies. The two branches of the ConvLSTM network initially have a similar data structure. However, after merging, the time steps are no longer a single dimension of the data structure ([samples, time steps, features]). It is not certain if the model is able to match the ConvLSTM input timed matrices with the associated timed vectors of the (filed) flight points. Although the input data is carefully selected and structured, it can be argued that this research has depended too much on the capabilities of a deep-NN (extract patterns among unstructured data). Merging two different data sources with different structured time vectors might have inhibited the ConvLSTM network from functioning as expected.

VIII. Conclusion

The research objective is to improve the accuracy of medium- to long-term flight trajectory predictions by incorporating a model that encompasses the dynamics of the air traffic situation. The shift towards TBO and free flight reduce the dependency on (inefficient) standard routes while demanding an improved ability to predict 4D trajectories. To the best of my knowledge, this research is the first to incorporate air traffic dynamics features into a trajectory predictor using a deep LSTM-based NN that includes ConvLSTM layers.

The air traffic dynamics features are quantitatively represented on a spatiotemporal map. In the first phase of this research, the statistical analyses reveal that the traffic density, mean separation between flights, and heading variance have a significant monotonic correlation with horizontal track deviation. No statistically significant relation is found between the air traffic dynamics and the flight level deviation nor with flight delays.

The second phase involves incorporating the selected air traffic dynamics features into a novel 4D trajectory predictor. The trajectory predictor is a composite deep NN, merging a ConvLSTM-based network with a stacked-LSTM network. The model is able to predict the 4D flight trajectories with reasonable accuracy: 75% of the predictions have a CTE below 15 NM at 28 minutes of look-ahead time, with a median at roughly 8 NM. However, based on the evidence in this research, it is concluded that incorporating the air traffic dynamics in the current ConvLSTM-based NN does not lead to observable improvements to the accuracy of medium- to long-term 4D flight trajectory predictions. The effects of air traffic dynamics on the accuracy of trajectory prediction are non-observable as a consequence of three reasons combined. First, the adjustments in flight path and speed made by en-route flights are very small. Second, there are variations in model performance that cannot be ignored. Running very large data sets or a large number of repeated experiments is not feasible due to computational limits. Third, the filed flight points dominate the performance of the trajectory predictor.

The conclusion is based on the evidence collected from a sub-optimal composite ConvLSTM model and thus, have not been validated. Nevertheless, based on the evidence, it is likely that an improved version of the composite ConvLSTM with air traffic dynamics, especially traffic density, can improve the predictions of 4D flight trajectories. However, compared to a stacked-LSTM NN, the significant increase in computational effort that is required for an all-encapsulating composite ConvLSTM NN with a large variety of input data does not outweigh the suspected small gain in 4D trajectory prediction accuracy.

The gained knowledge to the ATM-domain is as follows. The proposed method in this study is independent from standardized routes and controller behavior or procedures, which makes it especially suitable for free flight trajectory prediction. It is shown that the surrounding air traffic dynamics are of some, but minimal, influence to the 4D trajectories of individual flights. This means that pilots and in-part ATCOs may not be as sensitive to the surrounding air-traffic situation as previously expected. This conclusion only holds for en-route flights in FRA. This, in fact, encourages the adaptation of free flight because insensitivity to surrounding air traffic dynamics also means that excessive ATC interventions and procedures may be obsolete. This conclusion however, must be validated in follow-up research.

Improvements to the model design, more extensive pre-filtering, and a better understanding of air traffic dynamics are required to successfully implement the LSTM-based deep NN for trajectory prediction under TBO-enabled regions.

IX. Reccomendations

In response to this research, several recommendations are proposed for future studies and as improvements to the deep learning model.

To improve the efficiency and performance of the LSTM-based model, a few recommendations are proposed. First, the improvements to the stacked-LSTM model can be translated to the composite ConvLSTM model, as discussed in Section VII-A. Next, it is recommended apply more filtering of flights and split the LSTM NN up in several specialized parts. This approach resembles clustering, which is applied in multiple studies on data-driven trajectory predictions. However, instead of clustering standard routes, it proposed to define the clusters based on the flight path pattern. First, the flights could be passed through a classifier to distinguish between flights that adhere to the flight plan or not. Second, another classifier could use the filed flight plan to distinguish between flight patterns: straight flight paths, right- and left-hand turning flights above a threshold, and irregular flight paths including significant vertical manoeuvres. Alternatively, clustering is also possible based on feature distribution, such as DBSCAN. One can analyze and process these eight batches of flights separately, which will also allow longer time-frames -in excess of one month- to be considered. For flights with straight paths that adhere to the filed path, it is probably best to simply follow the filed flight path. For flights that do not adhere to the flight plan, the input data can be passed through two parallel networks: a stacked LSTM network with the flight path and information data; and in parallel, a ConvLSTM network to incorporate air traffic dynamics information. Although the use of ConvLSTM layers is a proven technique to deal with spatiotemporal data structures, it is expected that the time vector discrepancies limit the usability when merging various data sources. It is therefore not recommended to merge the two networks but allow each network to process the unique input data, and make separate predictions. Separate predictions can be combined through a simple NN in order to obtain a single prediction. This will also allow for increased explainability of the model and features. Finally, validation of the deep learning model entails testing the model under various realistic environmental conditions, such as region of application and time-frames. Moreover, a comparison of the prediction accuracy to an exiting trajectory predictor (mostly aircraft performance models) would complete the validation.

In my opinion, understanding the effects of the surrounding air traffic dynamics on individual flights will become increasingly important due to the rise in TBO and free flight. This study has identified some weak relationships. However, it is recommended to study these relationships in-depth and produce validated conclusions. Some air traffic dynamics models that are recommended to be studied in further detail are the Lyapunov exponents, convergence-divergence rates (although intrinsic to the Lyapunov exponent), and conflict predictions. It is suitable to conduct human-in-the-loop experiments by subjecting pilots to various air traffic dynamics scenarios. If the conclusion holds that en-route flights are largely insensitive to surrounding air traffic and it is better understood how air traffic dynamics affect pilot decision making, it may allow a relaxation of ATC interventions and speed-up the adaptation of free flight. This development could be disruptive to the ATM domain. It is therefore strongly recommended that further research is done in this field.

References

[1] T. Radisic, D. Novak, and B. Juricic, "Reduction of air traffic complexity using trajectory-based operations and validation of novel complexity indicators," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 11, pp. 3038–3048, 2017.

[2] B. Hilburn, "Cognitive Complexity in Air Traffic Control - A Literature Review," Eurocontrol Experimental Centre, Tech. Rep., 2004.

[3] P. Andraši, T. Radišić, D. Novak, and B. Juričić, "Subjective air traffic complexity estimation using artificial neural networks," *Science in Traffic & Transportation*, vol. 31, no. 4, pp. 377–386, 2019.

[4] I. V. Laudeman, S. G. Shelden, R. Branstrom, and C. L. Brasil, "Dynamic density: An air traffic management metric," NASA Ames Research Center, Tech. Rep. April 1998, 1998. [Online]. Available: https://ntrs.nasa.gov/search.jsp?R=19980210764 http://tinyurl.com/aqslsy9

[5] B. Számel, I. Mudra, and G. Szabó, "Applying Airspace Capacity Estimation Models to the Airspace of Hungary," *Periodica Polytechnica Transportation Engineering*, vol. 43, no. 3, pp. 120–128, 2015. [Online]. Available: https://pp.bme.hu/tr/article/view/7512/6773

[6] L. Piroddi and M. Prandini, "A geometric approach to air traffic complexity evaluation for strategic trajectory management," *Proceedings of the IEEE Conference on Decision and Control*, no. October 2014, pp. 2075–2080, 2010.

[7] M. Prandini, L. Piroddi, S. Puechmorel, and S. L. Brázdilová, "Toward air traffic complexity assessment in new generation air traffic management systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 809–818, 2011.

[8] E. Salaün, M. Gariel, A. E. Vela, and E. Feron, "Aircraft proximity maps based on data-driven flow modeling," *Journal of Guidance, Control, and Dynamics*, vol. 35, no. 2, pp. 563–577, 2012. [Online]. Available: http://arc.aiaa.org

[9] M. Prandini and J. Hu, "A probabilistic approach to air traffic complexity evaluation," *Proceedings of the IEEE Conference on Decision and Control*, no. March, pp. 5207–5212, 2009.

[10] M. Prandini, V. Putta, and J. Hu, "A probabilistic measure of air traffic complexity in three-dimensional airspace," *International Journal of Adaptive Control and Signal Processing*, vol. 21, no. February, pp. 731–744, 2010.

[11] S. Abdul, C. Borst, M. Mulder, and M. Van Paassen, "Measuring Sector Complexity: Solution Space-Based Method," *Advances in Air Navigation Services*, no. June 2015, pp. 10–34, 2012.

[12] D. Delahaye and S. Puechmorel, "Air traffic complexity: towards intrinsic metrics," *3rd USA/Europe Air Traffic Management R&D Seminar*, no. June, pp. 1–11, 2000. [Online]. Available: http://tinyurl.com/affmbc6

[13] H. Wang, Z. Song, and R. Wen, "Modeling air traffic situation complexity with a dynamic weighted network approach," *Journal of Advanced Transportation*, vol. 2018, 2018. [Online]. Available: https://doi.org/10.1155/2018/5254289

[14] D. Delahaye and S. Puechmorel, "4D Trajectories Complexity Metric Based on Lyapunov Exponents," 2011. [Online]. Available: https://hal-enac.archives-ouvertes.fr/hal-01319023

[15] G. Vouros, "Data-Driven Aircraft Trajectory Prediction Exploratory Research," SESAR Joing Undertaking, DART consortium, Tech. Rep., 2017. [Online]. Available: http://dart-research.eu/

[16] A. M. de Leege, M. M. van Paassen, and M. Mulder, "A machine learning approach to trajectory prediction," in *AIAA Guidance, Navigation, and Control (GNC) Conference*, 2013. [Online]. Available: http://arc.aiaa.org

[17] Z. Wang, M. Liang, and D. Delahaye, "A hybrid machine learning model for short-term estimated time of arrival prediction in terminal manoeuvring area," *Transportation Research Part C: Emerging Technologies*, vol. 95, no. January, pp. 280–294, 2018. [Online]. Available: https://doi.org/10.1016/j.trc.2018.07.019

[18] ——, "Short-term 4D trajectory prediction using machine learning methods," in *The 7th SESAR Innovation Days*. SESAR Joint Undertaking, 2017.

[19] S. Ayhan and H. Samet, "Aircraft trajectory prediction made easy with predictive analytics," *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, vol. 13-17-Augu, pp. 21–30, 2016.

[20] Y. Liu and M. Hansen, "Predicting Aircraft Trajectories: A Deep Generative Convolutional Recurrent Neural Networks Approach," pp. 1–24, 2018. [Online]. Available: http://arxiv.org/abs/1812.11670

[21] Z. Zhao, W. Zeng, Z. Quan, M. Chen, and Z. Yang, "Aircraft trajectory prediction using deep long short-term memory networks," in *The 19th COTA International Conference of Transportation Professionals*, no. January, 2019, pp. 124–135.

[22] X. Li, L. Peng, X. Yao, S. Cui, Y. Hu, C. You, and T. Chi, "Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation," *Environmental Pollution*, vol. 231, no. December, pp. 997–1004, 2017.

[23] R. Fu, Z. Zhang, and L. Li, “Using LSTM and GRU Neural Network Methods for Traffic Flow Prediction,” in *31st Youth Academic Annual Conference of Chinese Association of Automation*. Institute of Electrical and Electronics Engineers Inc., 2016, pp. 5–9.

[24] Z. Shi, M. Xu, Q. Pan, B. Yan, and H. Zhang, “LSTM-based Flight Trajectory Prediction,” *Proceedings of the International Joint Conference on Neural Networks*, vol. 2018-July, 2018.

[25] A. M. Hernández, E. J. Magaña, and A. G. Berna, “Data-driven aircraft trajectory predictions using ensemble meta-estimators,” in *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, vol. 2018-Septe. Institute of Electrical and Electronics Engineers Inc., 12 2018.

[26] C. A. Dek, “Predicting 4D Trajectories of Aircraft using Neural Networks and Gradient Boosting Machines.”

[27] S. Choi, Y. J. Kim, S. Briceno, and D. Mavris, “Prediction of weather-induced airline delays based on machine learning algorithms,” in *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, vol. 2016-Decem. Institute of Electrical and Electronics Engineers Inc., 12 2016.

[28] Y. Liu, M. Hansen, D. J. Lovell, M. O. Ball, and R. H. Smith, “Predicting Aircraft Trajectory Choice-A Nominal Route Approach,” *8th International Conference on Research in Air Transportation*, 2018.

[29] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, W.-C. Woo, and H. Kong Observatory, “Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting,” Hong Kong University of Science and Technology, Tech. Rep., 2015.

[30] Y. Pang, N. Xu, and Y. Liu, “Aircraft Trajectory Prediction using LSTM Neural Network with Embedded Convolutional Layer,” *Annual Conference of the PHM Society*, vol. 11, no. 1, 9 2019. [Online]. Available: https://papers.phmsociety.org/index.php/phmconf/article/view/849

[31] Eurocontrol, “Monthly Network Operations Report - Analysis September 2018,” Network Manager Directorate Eurocontrol, Tech. Rep. September, 2018.

[32] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 11 1997.

[33] C. Olah, “Understanding LSTM Networks,” 2015. [Online]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/

[34] F. A. Gers, J. Schmidhuber, and F. Cummins, “Learning to forget: Continual prediction with LSTM,” *Neural Computation*, vol. 12, no. 10, pp. 2451–2471, 2000.

Appendix

A. A Case of Overfitting

The purpose of this section is to present the results of an example experiment that suffers from extreme overfitting. The results of run 6 from Table VI are given. In Figure 14, the loss and accuracy function is given. It is apparent that the model trains very well on the training set, however this is not translated to the testing set.

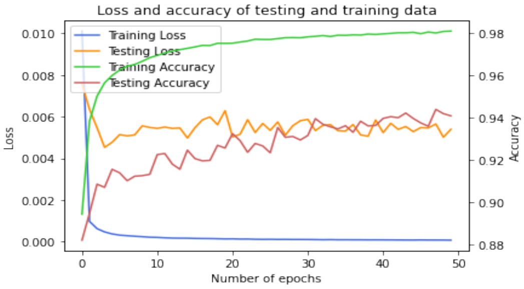


Fig. 14: The loss and accuracy function of run 6 from Table VI.

For illustrative purposes, the CTE, ATE and FL error are depicted in Figure 15, Figure 16, and Figure 17, respectively. The predictions are very poor and cannot be explained.

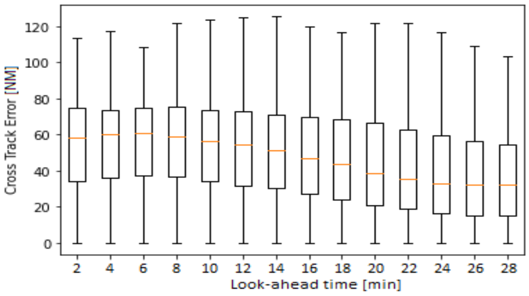


Fig. 15: The CTE of run 6 from Table VI.

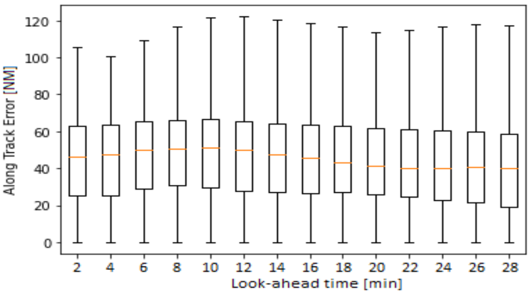


Fig. 16: The ATE of run 6 from Table VI.

B. A Visual Representation of the Initialization Error

Test 1a, 1b, and 2a suffer from the initialization error. This is deduced by observing the poor initial CTE, followed by an improvement of the CTE. However, it can also be observed well by simply looking at the trajectories on a map. In Figure 18, a few randomly selected flights are depicted, including the predictions. For nearly all the flights, the initial prediction lies far away from the last actual flight points. In most cases, the accuracy of the prediction actually increases slightly at increased look-ahead time. The expected reason for this increase is that the model has accumulated more sequential input information from the filed flight points.

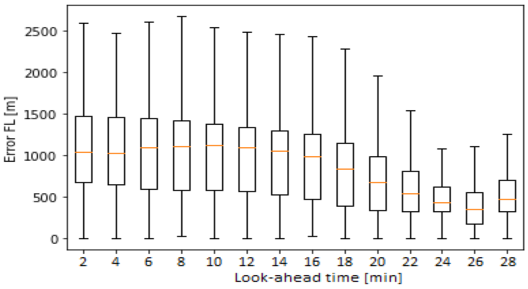


Fig. 17: The FL error of run 6 from Table VI.

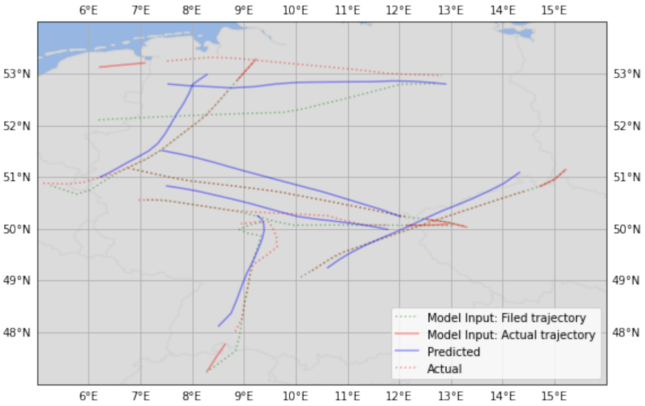


Fig. 18: A random selection of predicted trajectories at 28 minutes of look-ahead time, compared to the actual trajectories. This figure corresponds to run 10 in Table VI.

C. Results of the Unseen Air Traffic Dynamics Features

In the body of the paper, the results of *TD*, random grid values, and all-zero grid values are shown only. In this section, the results of the remaining air traffic dynamics features, *SDHdg* and *AvgDst* are presented.

1) Test 1a: The results of Test 1a correspond to the three day data set with whole day splits to separate the testing and training data. In Figure 19, the CTE remains practically constant with increased look-ahead time, whereas the CTE in Figure 20 gradually deteriorates. However, the CTE of *AvgDst* starts better but at long look-ahead times has slightly worse CTE. Despite the different distribution, the order of magnitude of the CTEs in both figures is similar to that of *TD* and the random grid values depicted in Figure 7a and Figure 7b, respectively.

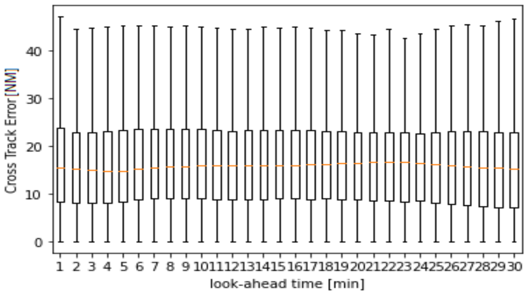


Fig. 19: Test 1a, run 2 in Table V: The CTE of the model with *SDHdg* feature values.

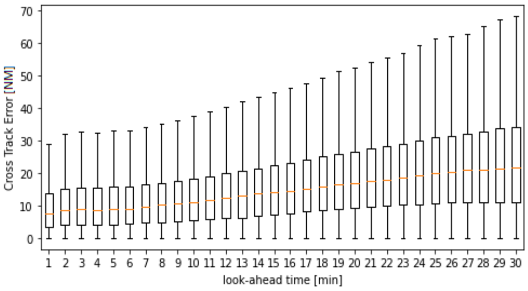


Fig. 20: Test 1a, run 3 in Table V: The CTE of the model with *AvgDst* feature values.

2) Test 1b: The results of Test 1b correspond to the seven day data set with randomized testing and training data sets. Figure 21 and Figure 22 present the CTEs of the runs with *SDHdg* and *AvgDst* grid values, respectively. Both models suffer from the initialization error. Generalizing, the accuracies of both predictions are comparable to the model with random grid values, seen in Figure 9b. The model with *SDHdg* grid vlaues is slightly better compared to the random grid values whereas the *AvgDst* grid values is slightly worse.

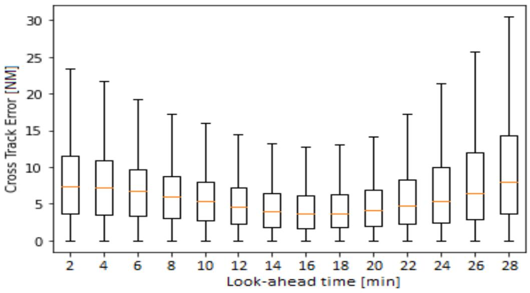


Fig. 21: Test 1b, run 8 in Table VI: The CTE of the model with *SDHdg* feature values.

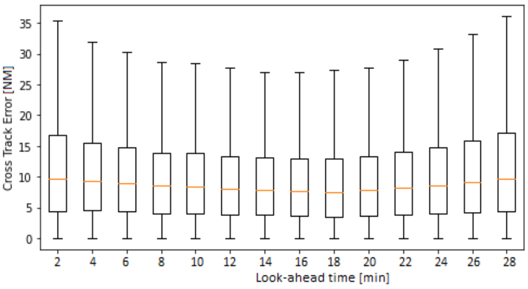


Fig. 22: Test 1b, run 9 in Table VI: The CTE of the model with *AvgDst* feature values.

D. Baseline Model Performance Variations

This section intends to illustrate the variations in baseline model performance due to model and suspected GPU backend library randomization. This section is based on test 2a, the results can be directly compared to Figure 10. All three results (including Figure 10) are repeated runs of identical models with identical data sets and an identical random seed. In Figure 23, the baseline model leads to very small CTEs across all look-ahead times. In Figure 24, the CTE is larger at all look-ahead times. Both models suffer from initialization error. The difference in performance between these runs is significant.

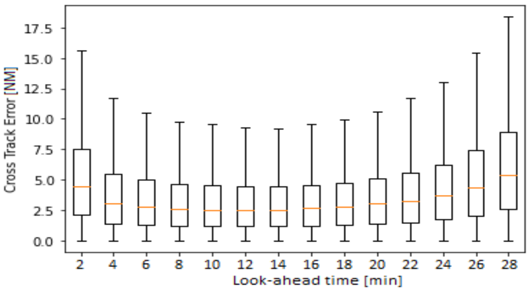


Fig. 23: Test 2a, an example baseline model run with good prediction performance.

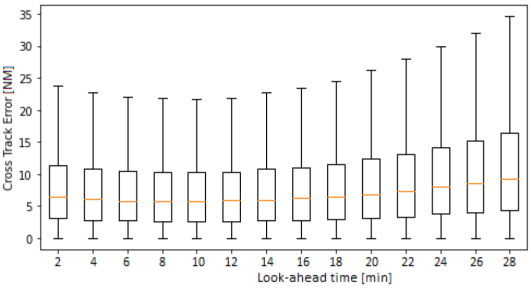


Fig. 24: Test 2a, an example baseline model run with worse prediction performance.

II

Preliminary Report

Summary

The arrival of Trajectory Based Operations in the next generation ATM system and increased pressure on efficiency of air transportation require improved trajectory prediction capabilities. This preliminary report is part of a thesis project to better understand the effects of the air traffic dynamics on individual trajectories and apply this knowledge to improve the performance of trajectory predictions. This two-step research will consist of both modeling the dynamics of the air traffic situation and analyzing the correlation of this model with trajectory changes by statistical analysis. Next, the selected features with the most significant correlation will be included in the trajectory prediction in an attempt to improve the performance of the predictor. This will be done using the Long Short-Term Memory Neural Network. The region of application will be one of the most congested en-route regions in Europe, the Karlsruhe Upper Area Control. This report includes a literature study on models of air traffic dynamics situations and data-driven trajectory prediction methods. From this study, conclusions are drawn about the most suitable models and method to analyse and predict the trajectories. The statistical analysis suggests that some features derived from the Dynamic Density air traffic complexity model are correlated with the horizontal track deviation. The correlations with delay and Flight Level deviations are non-significant. It is concluded that analysis of additional features that model the air traffic dynamics are required. Nevertheless, the most promising features are selected to be included in the Neural Network. The Long Short-Term Memory model is most suitable for the time series 4D trajectory prediction. This report gives a structured overview of the necessary steps that need to be taken to perform this prediction and subsequently validate the results. Lastly, the planning for the remaining phase of this thesis project is made.

List of Figures

List of Tables

Acronyms

ANSP	Air Navigation and Service Provider
APM	Aircraft Performance Model
ATC	Air Traffic Control
ATCO	Air Traffic Control Operator
ATFM	Air Traffic Flow Management
ATM	Air Traffic Management
DART	Data-driven AiRcraft Trajectory prediction research
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DCB	Demand and Capacity Balance
ETA	Estimated Time of Arrival
FRA	Free Route Airspace
GBM	Gradient Boosting Machines
GLM	Generalized Linear Model
HMM	Hidden Markov Model
KDE	Kernel Density Estimation
LSTM	Long Short-Term Memory
NN	Neural Network
PBM	Performance Based Methods
PCA	Principal Component Analysis
PCA	Principle Component Analysis
RMSE	Root Mean Square Error
RNN	Recurrent Neural Network
SVR	Support Vector Regression
TBO	Trajectory Based Operations

1 Introduction

Due to the ever growing demand for air transportation, the operational capabilities of *Air Traffic Management* (ATM) are reaching their ceiling in terms of capacity, efficiency, and cost effectiveness. The disruption of commercial air transportation from the COVID pandemic forced extra pressure on the efficiency and cost effectiveness of the ATM system.

In the past years, a shift is made towards *Trajectory Based Operations* (TBO). Interconnection, sharing of information, and the integration with decision support tools will allow optimized services for all ATM stakeholders. This integration, and thus dependability, of services can only be attained with high accuracy 4D-trajectory prediction. In other words, it is increasingly important to be able to predict the future path that an aircraft will fly, especially when given more freedom to choose ones own trajectory. TBO has been embraced and placed high on the agenda at both the Single European Sky program as well as with the Next Generation Air Transportation System project of the FAA. [7, 14]

There continues to be a discrepancy between the predicted trajectories and the actual flown trajectories, especially at medium- to long-term look-ahead times. The deviations are expected to come from ATC interventions or a change in aircraft intent to ensure optimum and safe operations. Research has been done on various factors that affect the trajectory such as the weather, aircraft intent, fixed (arrival) routes, fixed locations of way points, and aircraft types[8, 54]. However, the effects of surrounding air traffic and the dynamic behavior of this traffic has so far been under-explored in previous research as a contributing factor to individual trajectories. The assessment and impact of the air traffic complexity has been considered in various research related to *Air Traffic Controller* (ATC) workload modeling, which to a limited extend relates to air traffic dynamics. From this point onward, the term air traffic dynamics refers to the instantaneous and time-varying state of the air traffic inside a specified region as well as the inter-aircraft dynamical behavior. As air traffic and environmental data is more widely available and of higher fidelity, it is increasingly beneficial to apply data-driven techniques for trajectory prediction. One of the advantages of using data-driven techniques is that it can encapsulate more parameters and trends previously unforeseen with analytical methods. If, by including all relevant data, it can be shown that these interventions adhere to a patterns, big data analytics and machine learning techniques could reveal such a pattern.

The above explained research gap thus leads to the following research goals. The first goal of this research is to understand and identify relevant features of the air traffic dynamics that have a relationship with the trajectory. Secondly, these relevant features will be incorporated in existing data-driven trajectory predictors, thereby aiming to improve the performance of the trajectory prediction.

The research goals are specified further in Chapter 2 including the research objective and the research methodology. Next, Chapter 3 will cover the literature review. This chapter will include insights into previous research done on the air traffic dynamics, trajectory prediction, and introduce the input data that will be used. Based on the literature, the most promising air traffic dynamics model and trajectory prediction algorithm are proposed. Chapter 4 will give a overview of the experimental set-up for the trajectory prediction phase, including means to validate the results. Chapter 5 covers the preliminary analysis of the air traffic dynamics models. Although complete, the analysis has not yet been finalized. The identified limitations will be mitigated in the next phase of this research. Last, Chapter 6 gives a planning for the remainder of this thesis project.

2
Problem Statement

The purpose of this chapter is to present and elaborate upon the research objective. The research objective includes the research questions that this thesis attempts to answer. In Section 2.2, the methodology will be outlined. In this section it will become clear why and how this thesis is split into two major phases.

2.1. Research Objective

This project focuses on the medium- to long-term predictions of flight trajectories using a data-driven approach. The novelty of this project is to model the dynamics of the air traffic and incorporate this model in an attempt to improve flight trajectory predictions in the novel TBO environment. In other words, the main research objective of this thesis is:

> **To improve the accuracy of medium- to long-term flight trajectory predictions by incorporating a model that encompasses the dynamics of the air traffic situation.**

To achieve this objective, a series of sub-goals are defined. First, a study will be conducted on certain models and features that best describe the dynamics of air traffic situations. Next, a method must be devised to quantify and assess the selected models and features of the air traffic situation, either in real-time or as a forecast. Then, a data-driven trajectory prediction method is to be selected by assessing the suitability of the method in the chosen environment. An analysis will be conduced to asses the effect of the novel air traffic dynamics model and features on the performance of the trajectory predictor by a baseline comparison.

The main research question is as stated, followed by a series of sub-question that will provide necessary answers.

> To what extend do the air traffic dynamics have an effect on the individual trajectory in a free routing airspace and how can this knowledge be used to improve the predictability of aircraft trajectories?

In order to help answer the main research question as well as steer the conducted research methodology, a series of sub-questions are defined below that will need to be addressed.

1. What model derived from literature is most suitable to quantify the air traffic dynamics for the purpose of medium-to long-term trajectory predictions?
   - 1a. What are existing models that quantify the air traffic dynamics and complexity?
   - 1b. What features of the air traffic situation are expected have an impact on the flown trajectories?
   - 1c. What characteristics of the air traffic dynamics model make it suitable to a data-driven trajectory prediction application?

1d. How can the air traffic dynamic model be best visualized and represented for human intuitive understanding?

2. What is the correlation between the air traffic dynamics and the flown trajectories of individual aircraft?

   2a. What is the most suitable method for deriving the relationship between the chosen air traffic situation parameters and the flown trajectories?

   2b. What metrics of a trajectory will be taken into account to evaluate the flown trajectory?

   2c. Is there a statistically significant relation between the air traffic dynamics variables and the trajectory metric variables? If so, what is that relation?

3. Which existing data-driven trajectory prediction methods is best suited to incorporate the air traffic situation forecast?

   3a. What are the requirements and operating environment of the trajectory predictor and what makes it suitable?

   3b. What are suitable data-driven trajectory prediction methods and what can be said about the expected accuracy?

   3c. What is the output metric of the trajectory predictor?

4. What can be concluded about the performance of the extended trajectory predictor?

   4a. What are the means to assess the performance of the trajectory predictor?

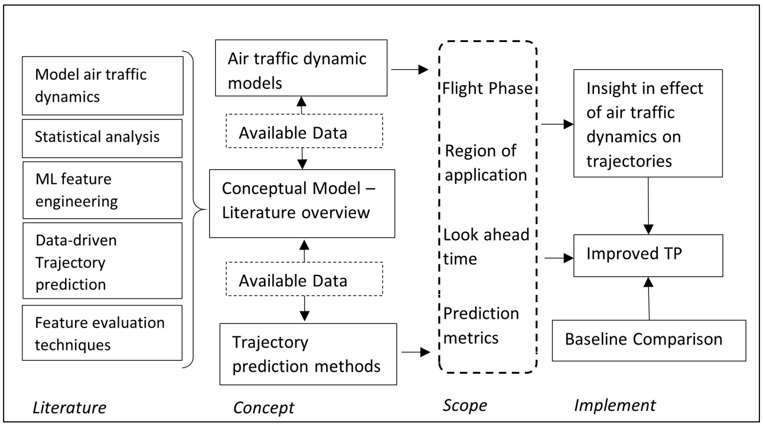
   4b. To what extend does the air traffic feature influence the performance of the trajectory prediction?

   4c. Is it possible to compare the same trajectory prediction with and without the air traffic dynamics as a baseline comparison?

   4d. What implications do the results have on the air traffic management in a practical context? This can be evaluated based on the throughput per unit of time or based on the (strategical) demand and capacity balance.

2.2. Research Methodology

This section will summarize the high-level method that will be followed during the research. Figure 2.1 provides a schematic overview of the conceptual model that provides the basis of the method. The four different parts are elaborated below.



Model air traffic dynamics
Statistical analysis
ML feature engineering
Data-driven Trajectory prediction
Feature evaluation techniques
Air traffic dynamic models
Available Data
Conceptual Model – Literature overview
Available Data
Trajectory prediction methods
Flight Phase
Region of application
Look ahead time
Prediction metrics
Insight in effect of air traffic dynamics on trajectories
Improved TP
Baseline Comparison
Literature
Concept
Scope
Implement

Figure 2.1: The conceptual model

The theoretical basis of the work consists of several parts. First, a collection of air traffic dynamics models derived from literature will be selected. This will answer research question 1. Next, the selection of models or features will be analyzed. The air traffic dynamics and complexity metrics will undergo feature engineering as well as a classical statistical analysis to evaluate if any relationships are significant. If this does not suffice, a method for feature evaluation will also be derived from literature. This will answer research question 2. Besides, a comparison of trajectory prediction methods will be made based on the available literature. The scope of the research will have to indicate whether it is sufficient to select one - most suitable- trajectory prediction technique or if multiple are needed for a reasonable comparison of results. This step will answer research question 3.

The scope of the conducted research will have implications on the extend of the research and must be very carefully chosen. The scope consists, among others, of the flight phase and region of application that is considered. The look-ahead time as well as the prediction output and metrics will be very decisive for the most suitable air traffic model. Prediction metrics are used to evaluate the output variables. The trajectory prediction can output a 4D trajectory which is a 3D individual trajectory with respect to time. However, this might not fit the scope of the research, in which the prediction will consists of at least a single time of arrival at the edge of the chosen region of application - a delay. Together, these results and choices will lead to a conclusion about the effect of the air traffic model on the trajectories. This will answer research question 4. A baseline comparison will be conducted. Lastly, the airspace capacity is dictated by the ATC capacity and required safe separation margins. By improving the trajectory prediction, it will be explored if the safe separation margins can be relaxed and thereby increase the airspace capacity. This will answer research question 4d. Following the initial stages of the research, including the literature study, a detailed work flow diagram was made and can be seen in Figure 2.2.

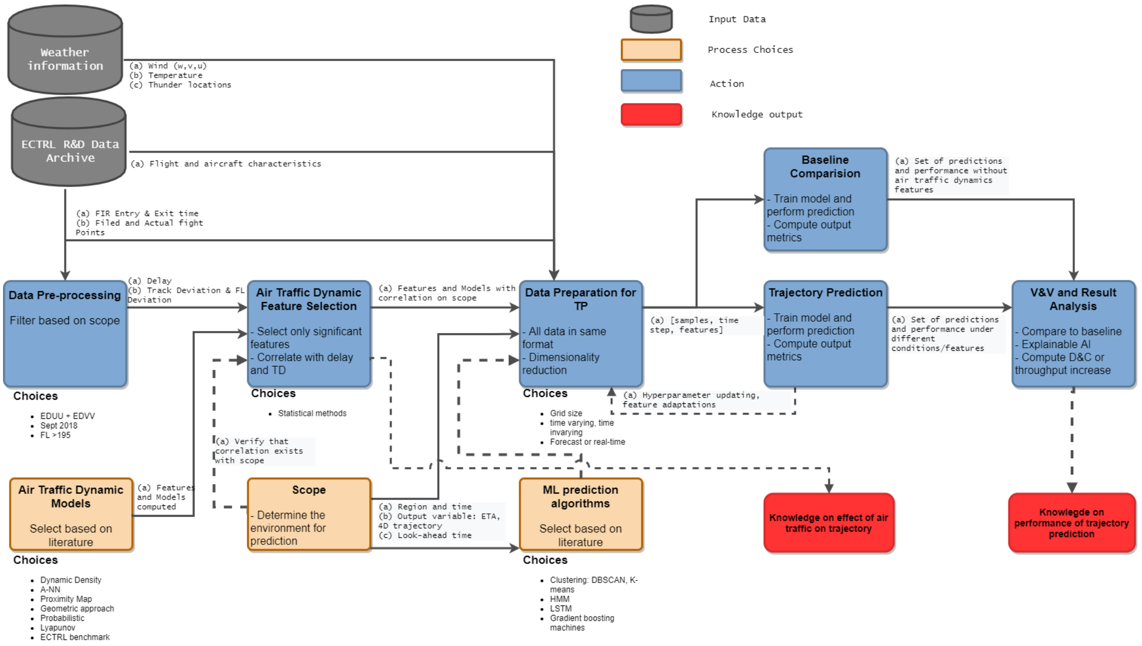

Input Data
Process Choices
Action
Knowledge output
Weather information
(a) Wind (w,v,u)
(b) Temperature
(c) Thunder locations
ECTRL R&D Data Archive
(a) Flight and aircraft characteristics
(a) FIR Entry & Exit time
(b) Filed and Actual fight Points
Data Pre-processing
Filter based on scope
Choices
EDUU + EDVV
Sept 2018
FL >195
(a) Delay
(b) Track Deviation & FL Deviation
Air Traffic Dynamic Feature Selection
- Select only significant features
- Correlate with delay and TD
Choices
Statistical methods
(a) Features and Models with correlation on scope
Data Preparation for TP
- All data in same format
- Dimensionality reduction
Choices
Grid size
time varying, time invarying
Forecast or real-time
(a) [samples, time step, features]
(a) Hyperparameter updating, feature adaptations
Baseline Comparision
- Train model and perform prediction
- Compute output metrics
(a) Set of predictions and performance without air traffic dynamics features
Trajectory Prediction
- Train model and perform prediction
- Compute output metrics
(a) Set of predictions and performance under different conditions/features
V&V and Result Analysis
- Compare to baseline
- Explainable AI
- Compute D&C or throughput increase
(a) Verify that correlation exists with scope
Air Traffic Dynamic Models
Select based on literature
Choices
Dynamic Density
A-NN
Proximity Map
Geometric approach
Probabilistic
Lyapunov
ECTRL benchmark
(a) Features and Models computed
Scope
- Determine the environment for prediction
(a) Region and time
(b) Output variable: ETA, 4D trajectory
(c) Look-ahead time
ML prediction algorithms
Select based on literature
Choices
Clustering: DBSCAN, K-means
HMM
LSTM
Gradient boosting machines
Knowledge on effect of air traffic on trajectory
Knowlegde on performance of trajectory prediction

Figure 2.2: Work Flow diagram of entire research project.

3 Literature Review

This chapter gives an overview of the related research that has been conducted on topics that relate to this Thesis. This includes the developments in Air Traffic Management in Section 3.1, a study about air traffic dynamics in Section 3.2, Trajectory prediction in Section 3.3. In Section 3.4, the data is introduced that will be used in this study. Section 4.1 and Section 4.2 bring together the studied content and proposes which models are most suitable for the application of this Thesis. This is covered in the next chapter as part of the research plan but relates strongly to the literature review.

3.1. Developments in Air Traffic Management

The operational capabilities of ATM are reaching the ceiling in terms of capacity, efficiency, and cost effectiveness. This has been a continuing challenge for many decades. Already in 1983 the Future Air Navigation System special committee of ICAO was created to make long-term recommendations on solution. The Single European Sky initiative and USA equivalent of NextGen are ongoing modernization projects both aiming to increase capacity, increase overall efficiency, increase safety, and reduce environmental impact. The international demand for air transportation has been greatly reduced as a result of the COVID-19 pandemic. The effects of this disruptive event is expected to suppress the demand for air transportation for several years. This has put even more pressure on the profit margins of all stakeholders part of the air transportation chain as well as the increased the call for efficiency. The areas of development for both projects span across all relevant domains and all flight phases. A cornerstone of both projects is the ATM concept of TBO.

Trajectory Based Operations

TBO places four dimensional trajectory information in the center of the ATM chain, demanding that all stages of the trajectory life-cycle, from planning to execution and amendments, are linked. In TBO the pilot and airliner have more freedom to choose and optimize the route, and are not strictly limited to waypoint-to-waypoint navigation upon ATM instruction. Interconnection, sharing of information and the integration with decision support tools will allow optimized services for all ATM stakeholders. This integration, and thus dependability, of services can only be attained with high accuracy (4D) trajectory prediction. A key element of TBO is the *Performance Based Navigation*, enabling aircraft to navigate along their optimal trajectories. Moving away from point-to-point navigation along NAVAIDS and procedures based on standardized routes and towards more flexible PBO can increase the efficiency, capacity, and safety. Figure 3.1 shows the evolution of en-route navigation and operation from ground-based navigation aids to *Area Navigation* (RNAV) and *Required Navigation Performance*(RNP).

Moreover, ATC capacity, procedures, and a lack of shared information places significant restrictions on the demand and capacity balance. Accurate global navigation and improved decision support tools for ATC have allowed an increased shift to RNAV operations, providing timely alerts when an aircraft deviated from its assigned route. Naturally, this places an increased requirement on the accuracy to be able to predict the position of an aircraft. PBN and operations are adjusted based on the environment and predicted air traffic demand. [14] A descriptive overview of the benefits of increased accuracy of trajectory prediction in all flight

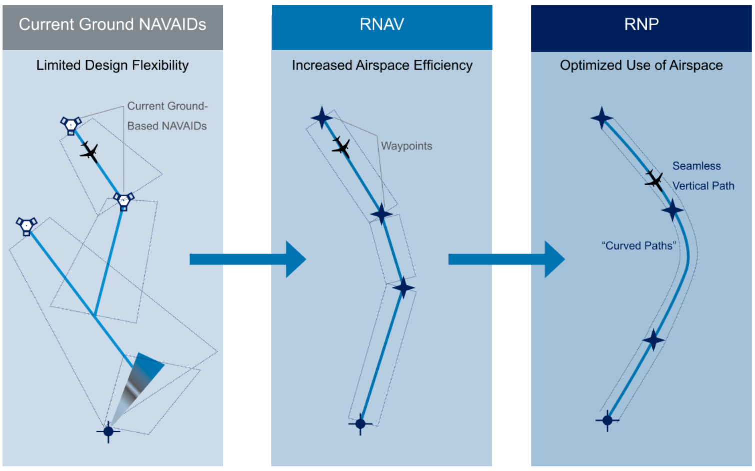

Current Ground NAVAIDs
Limited Design Flexibility
Current Ground-Based NAVAIDs
RNAV
Increased Airspace Efficiency
Waypoints
RNP
Optimized Use of Airspace
Seamless Vertical Path
"Curved Paths"

Figure 3.1: Evolution of en-route navigation and operations. [14]

phases is shown in Figure 3.2. The required alterations to operations cannot change overnight. Eurocontrol works with so called *Free Route Airspace* (FRA).

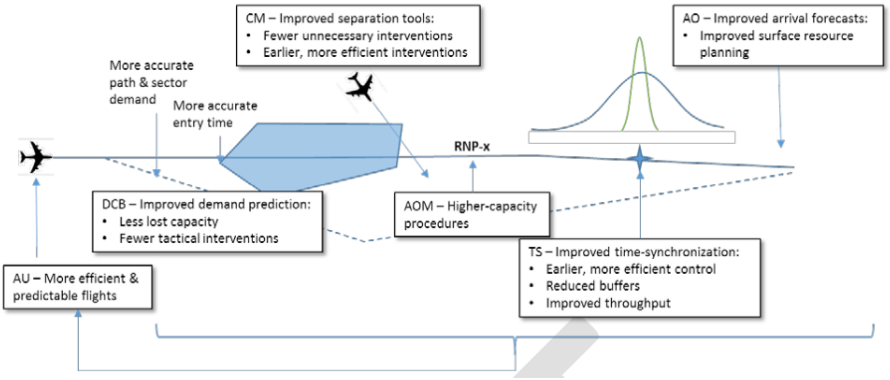

CM – Improved separation tools:
• Fewer unnecessary interventions
• Earlier, more efficient interventions
AO – Improved arrival forecasts:
• Improved surface resource planning
More accurate path & sector demand
More accurate entry time
RNP-x
DCB – Improved demand prediction:
• Less lost capacity
• Fewer tactical interventions
AOM – Higher-capacity procedures
TS – Improved time-synchronization:
• Earlier, more efficient control
• Reduced buffers
• Improved throughput
AU – More efficient & predictable flights

Figure 3.2: The benefits of improved trajectory prediction in all flight phases. [18]

Free Route Airspace

FRA are specified volumes of upper airspace that support the concept of operations in which a user has the freedom to plan a route between defined entry and exit waypoints. Flights remain subject to ATC and, depending on airspace availability, routing is possible via intermediate waypoints. The FRA can be considered a operationalisation of RNAV, in which aspects of a TBO environment are realised. Studies by Eurocontrol show that the workload of controllers is decreased as a result of FRA. In line with the objective of this study, it is desirable to study the air traffic dynamics in a sector that has some degree of autonomy. Otherwise, there exists a risk that dynamical effects are entirely dictated or possibly nullified by the strict procedural ATC commands. It is therefore relevant to consider airspace that supports FRA in order to study the dynamical inter-aircraft behavior. Figure 3.3 shows the implementation of FRA in European Upper Control Area airspace in 2018. [1]

3.2. Air Traffic Dynamics

3.2.1. Air Traffic Dynamical Models and its Applications

Literature related to the modelling of air traffic dynamics can be roughly divided into two categories, depending on the phase of flight under consideration: (pre-)tactical or in-flight. The tactical flight planning phase

[1] https://www.eurocontrol.int/concept/free-route-airspace

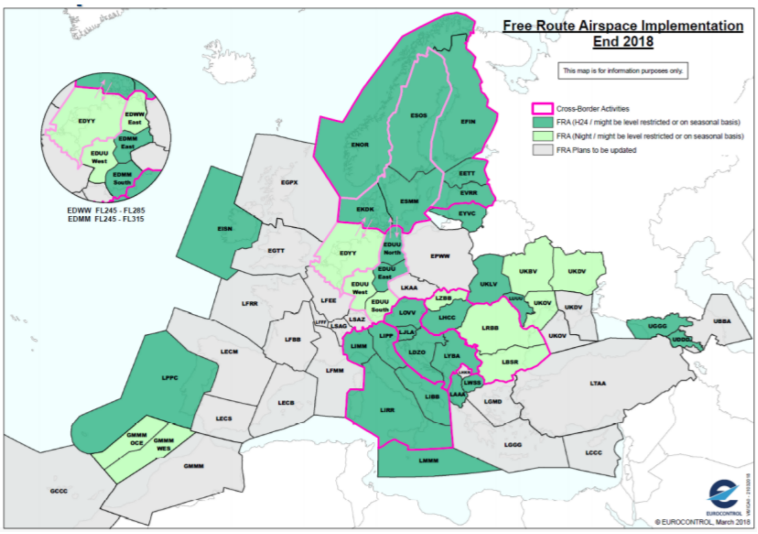


Figure 3.3: FRA implementation in the various Upper Control Area's in 2018.

refers to hours or minutes prior to departure. During this flight phase, the so called Demand and Capacity Balance (DCB) must be solved to try and accommodate all the departing or incoming flights. This includes en-route in a sector, whose capacity can vary during a day. It is important for the *Air Navigation and Service Provider*(ANSP) to know how many flights to expect in a sector prior to arrival in that sector. Any capacity issues will then delay the aircraft from departing or require alternate trajectories. [18] In this process, the Air Traffic Flow Management, requires air traffic dynamical models to predict what the airspace demand will be and how this varies over time. The in-flight phase, so the execution of the trajectory, relates to a different type of air traffic dynamics model. This type of air traffic dynamics model has been explored for air traffic complexity and workload modelling. Quantifying how complex a sector is can help ANSP predict the workload of controllers and make tactical decision on how to 'simplify' the traffic distribution to relieve the *Air Traffic Control Operator* (ATCO) workload. Thus, this type of air traffic dynamics (or complexity) modelling is mostly for shorter look-ahead times and individual trajectories instead of solving the DCB in a whole sector.

In fact, it is the intent of pilots and ATCOs that causes track deviations from planned trajectories. This intent is more likely to be influenced by air traffic complexity than by the DCB. Moreover, a change to the DCB might cause the publishing of a new filed flight plan, so it won't cause measurable track deviations. For these reasons, this literature study will focus on air traffic complexity and workload modeling in order to quantify the air traffic dynamics.

3.2.2. Air Traffic Complexity and Workload Modelling

The modelling of air traffic complexity has, in the early years of ATM, been done with the purpose of modelling the ATCO workload. For readability, a ATCO is from this point onward referred to as controller. A quantitative assessment of the cognitive complexity was needed to control and understand sector capacity as well as make progress in the automation of ATM. [5] However, the subjective controller workload is highly complex and includes qualitative as well as quantitative metrics. Mathematically, it thus seems near to impossible to reconstruct a perfect model of controller workload without sacrificing either the qualitative or quantitative model accuracy. Although this thesis is not focused on controller actions, understanding the previous work in these air traffic complexity metrics from the controller workload point of view is expected to contribute to the understanding of air traffic dynamics.

No doubt, complexity does directly influence the perceived difficulty to maintain safe and effective air traffic flow and in turn have an effect on the airspace sector capacity. But, as Radisic agues, 'complexity is not a synonym for workload'.[38] Although a crucial factor for measuring controller workload, the relationship between complexity and workload is mediated by several factors, including equipment quality, individual controller differences, controller cognitive strategies, and ATC procedures. [5, 38] It cannot be assumed that controller workload is analogous for ATC complexity, which is especially true for a TBO environment. For this study, **complexity is defined as the traffic factors that impact the level of difficulty of the ATC task, disregarding internal system, procedural, and airspace factors.**

Dynamic Density

The NASA was first to propose *Dynamic Density* (DD). This metric is composed of traffic density and traffic complexity, which constitutes of various weighted complexity factors. The purpose was to measure the air traffic controller workload, with real-time operational functionality. It is a weighted linear DD function, initially including eight air traffic complexity terms in addition to air traffic density. Initially, the proposal included controller intent, but this was operationally not feasible. The linear equation for dynamic density is as follows:

$$DD = \sum_{i=1}^{n} W_i TC_i + TD \tag{3.1}$$

where W_i is the weight for each ith factor, TC_i is the ith air traffic complexity factor, and TD is the traffic density. [23] The weights were determined experimentally, and it was concluded that DD is a robust method to capture the air traffic complexity and substantial amount (22%)of variance in controller activity that would have not been accounted for by traffic density alone. [45] Over time, three more studies have been performed which led to three more sets of air traffic complexity factors, totalling up to forty unique factors. See Appendix A for the full set of unique factors. Various sets of factors have been shown to perform well depending on the specific sector, without any set of factors clearly outperforming the rest.[20] This suggests that the method of weighted linear density functions and especially the factor weights are very sector dependent as well as subjected to highly subjective controller workload ratings. This leads to situations in which the factor weights are too sector specific, much like overfitting a network, and thus it cannot be validated for a more general environment.

Masalonis et al. [32] suggested a reduced model that consists of 12 factors from the original 40 and made an attempt to identify and distinguish between subjective complexity and useful metrics for real-time Traffic Flow Management decision support. Again, correlation was found with the set of factors, but was also specific for each observed sector. In this paper, it was proposed to include a real-time workload prediction score per sector. This can then be used to make better informed decisions about rerouting. This is a interesting approach and if implemented would not only help improve the capacity balance, it would make the (re)routing choices predictable.

Artificial Neural Networks

A more novel approach to modelling controller workload, but with an effectively similar working principle to the linear regression of DD, is the use of *Artificial Neural Networks* (ANN). This is done by Andrasi et al. [3] and Szamel et al. [46] By feeding a large variety of complexity factors into a standard multilayer perceptron, which is a universal function approximator, the network will output a estimate of the complexity. Interestingly, the frequency of use for each complexity factor was recorded, giving insight into the weight that each factor affects the final complexity. The feature importance and performance is very similar to the DD. The main advantage over DD is that the input data does not have to be filtered based on the concept of operations, which is the case when comparing conventional route based and trajectory based operations. Also, the model can easily be adapted to suit a different output metric, such as delay or trajectory deviation. This can be a useful in practice, in which many different combinations of sectors, input factors, and outputs metrics can be studied without very heavy labor intensive data pre-processing phase.

Input-Output

A controller focused air traffic complexity model that is worth mentioning is the input-output approach by Lee et al. [24] This method considers the minimum control activity needed to accept an aircraft entering the sector to maintain a conflict-free situation. This scalar information can potentially be used a a factor to present complexity. A major disadvantages is that this method is very much focused and uniquely applicable to controller workload, which is not the primary focus of this study. Moreover, this paper has not done proper validation, so the performance cannot be assessed reasonably. However, the concept of representing complexity in a map based on the needed trajectory corrections need not be overlooked.

3.2.3. Free-routing and Decentralized Complexity Modelling

Controller workload centered methods all experience the discrepancy caused by the subjectivity of ATC workload which introduces an unreliability from the raters or, if the method is control independent, have a difficult time verifying the reliability of the workload indicator. Moreover, it is continually questioned what the implications of ATC workload mean for the sector capacity, traffic flow, and safety. This raises the question on what complexity is in the context of ATC workload.

With the gradual increase of aircraft on-board autonomy and self-separation with the implementation of TBO, a renewed push is made to model the air traffic dynamics and complexity without the controller workload in the loop at all. This will lead towards a partially decentralized control scheme for ATM. As described by Piroddi et al, the next generation ATM complexity evaluation will support the functionality of on-board trajectory prediction. Optimizing the effectiveness of the flight could include high complexity zones, potentially requiring an entering aircraft too many tactical manoeuvres to pass them through. [35] This section will cover a few of the methods that make such an attempt.

Proximity maps

A very different approach is that taken by Salaun et al. [43] In this approach, 'three-dimensional aircraft proximity maps that evaluate the future probability of presence of aircraft at any given point of the airspace' are introduced. [43] Unlike the previous two methods, this method is concerned with medium- and long-term horizons of 30 minutes or more for controller and TFM applications. The *presence maps* indicate regions of higher aircraft density and have potential to be of value in assessing the sector air traffic dynamic. This way, the higher density regions are clustered and not generalized in an entire sector, which is expected to help identify trajectory patterns within a sector. A downside is that the proximity maps are based on clustering of recurring trajectories, while the application of this project is concerned with TBO. This is a potential pitfall because within a TBO sector, aircraft do not adhere to strict reoccurring routes. However, the idea of proximity maps in combination with congestion or weather patterns can be used to study the effects of certain factors on the air traffic system, visualized and quantified by the proximity maps. Figure 3.4 shows the clustering of flow, represented by a distribution with respect to the centroid.

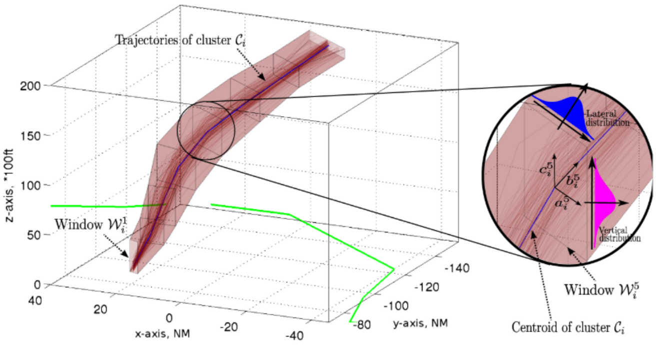


Figure 3.4: A 3D representations of a cluster of an ascending flow, used to generate proximity maps [43]

Trajectory Based Operations Complexity Indicators

Radisic et al. [38] questioned if the complexity factors that are applied in previous linear models, as explained in Section 3.2.2, during en-route based operations are also applicable to TBO. It was found that not only the inclusion of TBO trajectories significantly reduced controller subjective complexity, the predictive linear model using 20 standard en-route complexity factors did not perform well.(increasingly worse as fraction of TBO aircraft increased) This is because the TBO trajectories are 'strategically deconflicted' [38] and thus do not adhere to traditional flight patterns that have been studied in the past. These standard factors relate to those mentioned in Section 3.2.2. Multiple stepwise linear regression analysis was performed with additional TBO specific factors and was used to verify the TBO-specific factors. Listed in order of importance, 6 factors were identified to most accurately represent ATC complexity: "Number of aircraft, number of conflicts between conventional aircraft and aircraft flying according to TBO (aggregated over 600 seconds) (TBO-specific), fraction of aircraft in climb or descent, number of aircraft near sector boundary (<10 NM), fraction of TBO aircraft (TBO-specific), and number of aircraft pairs at 3D Euclidean distance less than 5 NM. This finding can be valuable in understanding the air traffic dynamics under TBO environment.

Probabilistic Approach

Similarly to the proximity maps, Prandini et al. [36] introduces a conflict map, but this method is based on the probability that an aircraft with reach a certain point in the section at a certain time. The current state and aircraft intent determine the future position, taking into account the uncertainty. This relies in part on trajectory prediction. The complexity maps is thus dynamic leading to the capability of trajectory prediction and moreover being able to identify regions with a "limited inter-aircraft maneuverability space". [36] A novelty, one that is useful for trajectory prediction, is that this approach allows the evaluation of complexity along a trajectory of a single aircraft, meaning that this can be done on-board per aircraft. However, this application was only successful in the case that aircraft are following a fixed heading and velocity trajectory. It is a promising method, but the implementation has not yet deemed feasible, especially in real-world cases.

Geometric Approach

Self-separation, on-board routing, and increased autonomy would not be computationally viable if each aircraft has to continuously evaluate the air traffic dynamics of the entire sector. A proposed solution consists of each aircraft to compute its own zone of influence and map any possible conflicts with other aircraft. [35] The zone of influence is a envelope of possible motions that leads to the set of possible locations reachable by an aircraft from its intended trajectory. "Complexity is then related to the presence and magnitude of intersections between influence zones of different aircraft." [35] It has been shown that hundreds of aircraft can be handle without excessive computational load. By approximating the influence zones with a polyhedron it is theoretically also possible to include other time varying no-fly zones, such as weather formations or restricted airspace. This has not been explored but falls within the realm of possibilities using this method of (geometrical) complexity. The advantage of this method is that this geometric approach relates complexity to the range of possible solutions, irrespective of context. This is how pilots and controllers alike will solve conflicts. If then, this geometric approach can be translated into a useful metric, it might be a good quantitative approximation that is related to the conflict resolution done by airspace users and thus be a good factor to predict the conflict resolution and thus the trajectory.

A similar approach put to practical use is the Solution Space Diagram (SSD), in which the SSD is thought of as a measure for sector complexity. [41] A schematic overview of how a solution space is defined is given in Figure 3.5. This approach however is intended to measure sector complexity for controller workload for short to medium time ranges. [47] Nevertheless, the concept can be adapted and applied in the relevant context.

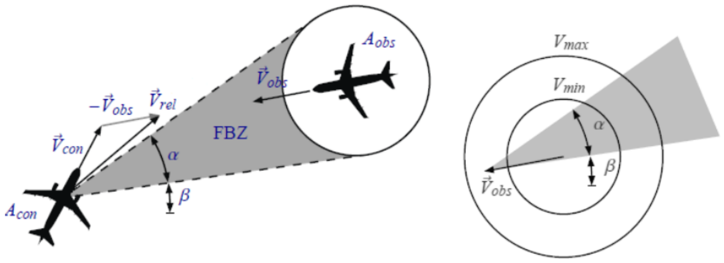


Figure 3.5: A schematic overview of a two aircraft condition, where the Forbidden Beam Zone is translated to a Solution Space. [41]

To reduce the subjectivity error, Delahaye et al. [10] introduced a methods to model the intrinsic traffic disorder which capture complexity and improve the dynamic density model. This is a geometrical approach, in which the relative motion of aircraft represents the traffic disorder, in similar fashion as in done by the human brain. The traffic disorder is represented by a three-dimensional complexity coordinate, composed of three axes: density, divergence/convergence, and (in)sensitivity. The advantages of this factor is that it is point specific and does not generalize the entire sector into a single value.

Dynamic Weighted Network

A novel objective measurement of air traffic complexity is proposed by the dynamic weighted network approach [48]. This model considers the effects of airspace structure and traffic characteristics by measuring the spatial approaching rate of three types of relationships: aircraft-aircraft, aircraft-waypoint, aircraft-route segment. Each summation of complexity measurements are normalized and aggregated with weighted factors for each complexity factor. It is found that this complexity measurement correlates directly with traffic count, correlates with a time-shift to trajectory changes, and correlates directly with sector conflict rate. The advantage of this method is the independence from sector geometry, and relative ease of computation. Only the aircraft-aircraft relationship is useful if used in a objective weighting scheme due to the scope of this thesis. This complexity metric, based on between aircraft spatial approaching rates (convergence), can supplement other complexity measurements. The between-aircraft complexity is given in Equation (3.2):

$$C^{A}_{i,j}(t) = \begin{cases} \left(\frac{1}{E_{i,j}(t)}\right)^{1+\beta_A {V^A}_{i,j}(t)} & ,E_{i,j}(t) \geq 1 \\ \left(\frac{1}{E_{i,j}(t)}\right)^{1-\beta_A {V^A}_{i,j}(t)} & ,E_{i,j}(t) < 1, \end{cases} \tag{3.2}$$

Where $E_{i,j}(t)$ is the ellipsoid distance between aircraft i and j at time t. $V^{A}_{i,j}(t)$ is the between-aircraft ellipsoid distance changing rate, see Equation (3.3). When $V^{A}_{i,j}(t)$ is larger than 0, the traffic situation is diverging. β_A is the adjustment coefficient for between-aircraft spatial proximity, which is not further specified.

$$V^{A}_{i,j}(t) = \frac{E_{i,j}(t) - E_{i,j}(t-1)}{E_{i,j}(t-1)} \tag{3.3}$$

Subsequently, the complexity values are then added for all aircraft pairs inside a selected region.

Lyapunov Exponents

Perhaps the most significant work done on modelling the intrinsic complexity of the air traffic is by Delahaye et al. [11], by pure dynamical systems modeling. The so called Lyapunov exponents are a traffic disorder metric that measure the sensitivity to initial conditions of the dynamical system. In words, Delahaye et al. explain this concept as follows: "The Lyapunov exponent map determines the area where the underlying dynamical system is organized. It identifies the places where the relative distances between aircraft do not change with time (low real value) and the ones where such distance change a lot (high real value)." [11] Figure 3.6 displays a Lyapunov Exponent map.

The inputs to this model are sets of trajectories, composed of positions and speed vectors. This method provides a very intrinsic and objective measures of complexity that is completely controller and sector geometry independent, can be represented on a map and the look ahead time is irrespective to the working method.

Eurocontrol's Complexity Metrics for ANSP Benchmarking Analysis

Both the ATC controllers and the pilot do not rely on advanced air traffic prediction models to forecast future state of the airspace in order to optimize the near-future trajectory. As such, the information that dictates the trajectory should be basic and intuitive, similar to basic human spatial awareness but also from the perspective of each individual aircraft. This is in part the reason why it is preferred to determine the complexity at spatiotemporal level on a map instead of a single scalar for the whole sector. Complexity Metrics for ANSP Benchmarking Analysis provides a method for determining the complexity based on each aircraft level of interactions with surrounding aircraft: "Interactions express the fact that it is the presence of several aircraft in the same area at the same time that generates complexity, particularly if those aircraft are in different flight

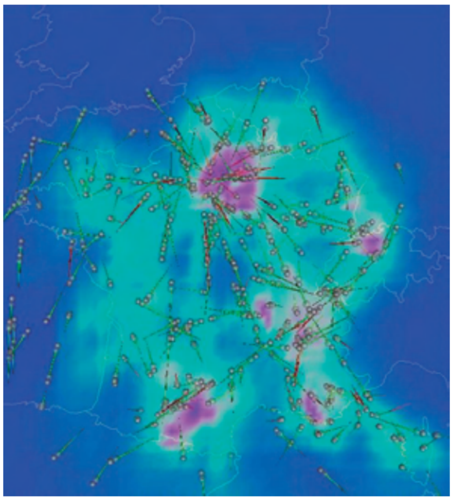


Figure 3.6: The results of the Lyapunov Exponent complexity of French Airspace, the red areas are regions of higher complexity [11]

phases, have different headings and/or different performances.".[2] The simplistic metric can easily be computed per aircraft through time as it passes through a sector. This is more a whole sector measure, but can be adjusted, aggregated per flight. This will allow for individual differences in the air traffic to make a valuable contribution to the regression model complexity indicators. The four factors are as follows, where each factor is defined as the ratio between the hours of interactions and flight hours:[2]

- Adjusted density - An interaction is defined as the simultaneous presence of two aircraft in a cell of 20x20 NM and 3000ft in height.
- Potential vertical interactions - Captures the potential interactions between climbing, cruising and descending aircraft.
- Potential horizontal interactions - Provides a measure of the potential interactions based on the aircraft headings.
- Potential speed interactions - Assesses the potential interactions based on the aircraft speeds.

3.2.4. Metrics to Evaluate the Air Traffic Dynamical Models

The intent to determine the air traffic dynamics relate to the trajectory prediction in a TBO environment in which FRA and self-separation will play a crucial role in the next generation ATM. It is with the research objective in mind that requirements are imposed on the air traffic dynamics model. These requirements, the air traffic dynamics metrics, are imposed to evaluate the suitability to trajectory prediction applications, and are described as follows. The method to evaluate the different complexity assessment model is inspired by Prandini et al. [37]

Air Traffic Complexity

As elaborated upon in Section 3.2.2, all the aforementioned models are means to determine the air traffic complexity. Not all methods are equally suitable to model the air traffic complexity for the application of this research. As mentioned, workload is mediated by procedures and ATC equipment, so a workload score is not always proportional to the air traffic complexity. The complexity model needs to be route independent, which is relevant for FRA and en-route traffic. Lastly, some previous work have introduced novel, but very specific, air traffic complexity features. These features might provide unique insight but do not provide an all encapsulating air traffic complexity model.

Look-ahead time

The time horizon will dictate the suitability of each model. Short- to medium-term complexity models often rely on conflict detection. These are computed based on the propagation of aircraft state and intent information over a time horizon up to 15 minutes. Accuracy as well as computational effort do not scale well beyond this time horizon. Medium- to long-term complexity model are computed based on flight plan and trajectory optimization. Regions of high complexity need to be identified along the planned trajectory which require excessive tactical maneuvers. The chosen model needs to be suitable for the desired medium- to long-term look ahead time in line with the objective of this research.

Control effort independence

Some models are generated independently of controller workload, but the application is concerned with modeling or aiding the (ground) control effort needed to handle the complexity of the air traffic. This is less problematic compared to subjective controller workload as an input because the control activity in future ATM is shifted from ground to user, but will not disappear. If a highly assumptive control effort prediction is used in the prediction of future trajectory, it is near to impossible to find causal relationships between air traffic complexity features and performance of trajectory prediction.

Prandini et al. [37] has introduced the notion of *flexibility*, which could exclude any controller dependency. It is proposed that the extent to which non-conflicting trajectories are present can be expressed by a degree of flexibility. Thereby not suggesting any preferential control action. The notion of flexibility differs depending on the look-ahead time. The long-term application of flexibility is concerned with weather systems, no fly-zones, and regions of high air traffic complexity for example. My considering how a model measures this flexibility, the control effort independence is evaluated.

Sector Independence

Ideally, FRA operates in a sector-free context, using Functional Airspace Blocks. Currently, this is not yet viable due to surrounding non-FRA operations and thus mandatory entry and exit points are defined. However, next generation ATM is heading towards largely sector independent operations. This means that air traffic complexity models should not be intrinsically dependent on the sector boundaries and other relevant sector specific characteristics. This does pose a problem when considering the air traffic density of a sector, which lies at the core of most complexity models. However, clustering and grid-like representation of density can help isolate airspace and substitute the need for strict sector boundaries.

Output Form

Air traffic dynamics is both space- and time-dependent. By condensing the aggregated space and/or time information of the traffic situation, the traffic dynamics can be expressed. [37] The air traffic dynamics can thus be expressed as scalar values, possibly dependent on time or alternatively space, to a spatiotemporal complexity map. Scalar values of complexity along the current and intended trajectory could provide meaningful information. For the long-term applications, it could be more insightful to identify a region of high complexity which the aircraft is expected to avoid. These two means of expressing the air traffic dynamics are not mutually exclusive.

3.2.5. Review of Air Traffic Dynamical Models

This section aims to summarize and give a structured overview of the possible methods to determine the air traffic dynamics. Table 3.1 provides a overview for each air traffic complexity method, evaluated for the metrics as determined in Section 3.2.4.

The output form of all methods is either a scalar or a map. In the case of scalar complexity values, it is the grid or sector size that largely determines the value. As such, some components that compose the scalar complexity value, can also be represented by a, discrete map. Density, for example, can be computed for the entire sector, but also on a smaller scale for a few square kilometers. The choice of grid size is therefore very important.

Nevertheless, since the scalar complexity values are the sum of a weighted linear regression model, or network, is is possible to discretize the scalar values on a grid representation. A significant number of researchers tend toward a weighted network of individual complexity features. Each individual feature can then be tested for correlation with a (later chosen) metric. This will be of great value during the feature engineering phase, in which certain features are selected and adapted to best suit the predictive model.

The probabilistic approach and geometric approach has not yet been proven successful on a scale that is needed in this research. That is because for long-term applications, the propagation of each aircraft trajectory needed to compute the available maneuvering space and influence zone respectively has not been proven accurate. The Lyapunov Exponent provides a spatiotemporal complexity map that is expected to closely resemble the intrinsic complexity, without a controller nor sector dependency and has no major limitation regarding the look-ahead time. The Eurocontrol complexity metrics are well suited to complement a grid-like representation and add to the scalar complexity value.

It can be concluded that a well reasoned selection of scalar complexity features in a regressive model, similar to the DD, ANN, and TBO complexity indicators is a promising method to determine the air traffic complexity, to which the Eurocontrol complexity metrics and a measure of traffic convergence can be added. Furthermore, the Lyapunov Exponent is the more viable map-based complexity method that takes into account system dynamics.

For the application of this research simplicity is very important, for two reasons. In order to understand and be able to distinguish between different features, there is no point in aggregating very many features together. On top of this, a pilot nor controller have a mental model of such a complex model, so the relation to the tactical trajectory changes must be based on actionable information.

Metric	Input Form	Working Method	Complexity	Controller Dependence	Look-ahead	Sector Depedency	Output Form	Overlap with other methods
Dynamic Density	Scalar features from AC state and trajectory information	linear regression	route-specific factors can be omitted; weights are validated with controllers; relevant features can be 'handpicked'	Only validation	Instantaneous measure, can be extrapolated by TP	Complexity no, weights yes	Scalar	Free to add features
ANN	Scalar features from AC state and trajectory information, unfiltered	NN	route-specific factors can be omitted; controller independent; relevant features can be 'handpicked'	Only validation	Instantaneous measure, can be extrapolated by TP	Complexity no, weights yes	Scalar	Free to add features

Input-Output	Timed AC trajectories	Minimizing control activity for de-conflicting manoeuvre	Not route specific;
			for controller applications;
			evaluation needed per AC.
				Yes, model controller	short- to mid-term	Yes, different solution space	Map, per AC!	Not directly

Proximity Maps	Historical trajectory information	Clustering routes and congested regions	Route specific, complexity is mapped	no	medium-to long term, identify regions of congestion	Sector specific	Spatiotemporal map	Not directly
TBO Complexity	Scalar features from AC state and trajectory information	linear regression	Not route specific, validated with controllers, complexity is generalized	no	Instantaneous measure, can be extrapolated by TP	Sector specific	Scalar	Free to add features
Probabilistic Appraoch	State and intent of AC	Find regions with limited maneuverability space.	Not route specific, per AC evaluation along trajectory	no	short- to mid-term	no	conflict map, per AC	Not directly
Geometric Approach	State and intent of AC	Individual zone of influence, map conflicts	Not route specific, per AC evaluation of 'influence zone', based on intersection of influence zones	no	long-term	no	conflict map, per AC	Not directly
Dynamic Weighted Network	State and intent of all AC, sector characteristics	measure the spatial approaching rate of: AC-AC, AC-WYPT, AC-route .	Route specific, controller validated, per AC evaluation along trajectory	no	short- to mid-term	no, only route dependent	conflict map, per region	Not directly
Lyapunov Exponents	State of AC	Traffic disorder, sensitivity to init conditions	Not route specific, controller free, intrinsic	no	any	no	spatiotemporal map	Not directly
ECTRL metric for ANSP	State and intent of AC	Measure potential interactions in volume	Not route specific, controller free, simplistic	no	medium	no	Scalar or map	Free to add features

Table 3.1: Comparison of various air traffic complexity models

3.3. Trajectory Prediction

Trajectory prediction been bee performed by numerous methods. In order to understand the various approaches and be able to comment on the advantages, disadvantages, and applications, the working principle must be understood. In this literature overview, the working principle of any predictor will be primarily defined by two main categorizations. The first categorization will be based on the prediction method used to define a trajectory, either nominal, worst-case, or probabilistic. The second categorization refers to the model used to actually propagate the aircraft and environment states to estimate a trajectory or set of trajectories. Moreover, aircraft intent inference is relevant for this research and will be further elaborated upon.

3.3.1. Trajectory Prediction Methodologies

Three different methods of trajectory predictions are observed:

- **Nominal (Deterministic)**
  Nominal methods predict a single trajectory by propagating the observed aircraft and atmospheric states along a single trajectory. This approach does not consider any uncertainties in the measurements making it only suitable for very short look-ahead times in a specific phase of flight and cannot be done before flight initiation [19, 27, 53].

- **Worst-case** Worst-case methods consider the range of possible trajectories and, depending on the objective, consider the worst-case scenario for trajectory prediction. This is a highly conservative method and inherently inaccurate since the trajectory with the highest likelihood is not by definition the worst-case, thus introducing error.

- **Probabilistic** Probabilistic methods take into account the known uncertainties to model the possible changes to the trajectory inputs or outputs. This can yield both analytical or numerical solutions. Nominal and worst-case approaches can be translated into a probabilistic approach. A nominal trajectory prediction corresponds to a case in which an aircraft follows a maximum-likelihood trajectory with probability of one. A worst-case prediction corresponds to a worst-case scenario in which a set of trajectories has equal likelihood.

A visual representation of the three mentioned methods is given in Figure 3.7. This categorization of methods is most applicable to performance based trajectory prediction models, which is further elaborated upon in Section 3.3.2. Nominal trajectory predictions are inherently deterministic as it yields an exact, analytical solution. Generally speaking, this means the models used to calculate a trajectory are (analytical) performance based models. Most data-driven prediction models are inherently probabilistic because historical data or input data with a specific distribution are translated into a most likely numerical solution, whether this is achieved by a machine learning algorithm or a Monte-Carlo simulation.

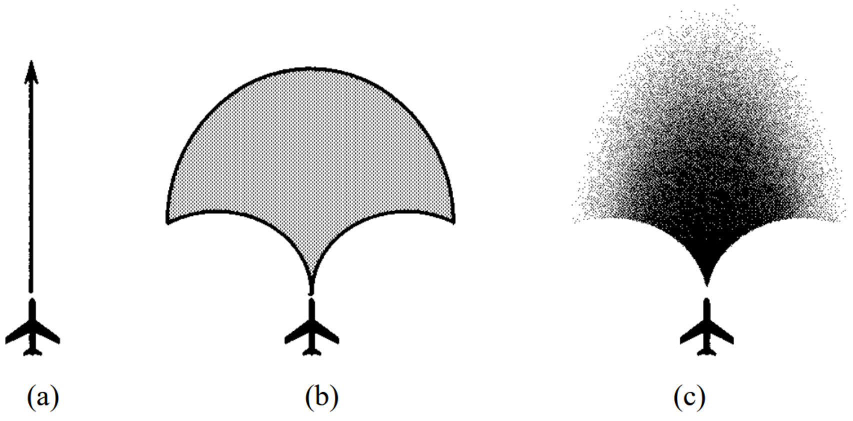


Figure 3.7: Visual representation of the nominal (a), worst-case (b), and probabilistic (c) trajectory prediction methods. [22]

3.3.2. Performance Based Trajectory Prediction Models

Performance Based Methods (PBM) rely on advanced aircraft models, called *Aircraft Performance Models* (APMs) as a function of the aircraft states and atmospheric conditions. This approach yields an analytically determined trajectory. The Base of Aircraft Data point-mass model is an example of a widely used performance model [1]. Point-mass models are a simplification of the aircraft dynamical system, considered reasonably accurate and commonly used in ATM research. [30] Common variables to APMs include: horizontal (X,Y) positions, altitude, heading angle, flight path angle, bank angle, true airspeed, Mach number speed, vertical speed, temporary level-offs, air temperature, lapse rate, wind.

APMs are a popular choice for many current applications of trajectory prediction. APMs are compatible with current ATM and ATC systems. This means changes to aircraft parameters can be easily applied to the existing system. For research or training purposes, these APMs give the flexibility of generating imaginary routes and scenarios while maintaining relatively accurate aircraft behavior. Missing or previously unseen information, such as geographic positions or routes, do therefore not limit its use. The look-ahead time however is limited to several minutes since the APM propogate current states and cannot predict a route. Intent information is necessary to predict with a longer look-ahead time. Moreover, nominal performance based methods are inherently inaccurate because these methods apply a deterministic model to a stochastic process [15]. The Base of Aircraft Data is a very accurate aircraft performance model, but a analytical approach will introduce intrinsic errors that produce deviation between the predicted and actual trajectory.

Note that the use of PBMs does not mean that probabilistic solutions to trajectory predictions are excluded. It simply refers to the dynamical aircraft model that is used to model aircraft behavior. For example, Monte-Carlo simulations to map the aircraft modeling discrepancies or weather forecast uncertainty allow for a Bayesian estimation problem to provide numerical solutions to aircraft trajectory prediction. Alternatively, Rudnyk et al. applied distribution functions to the various model inputs to model the prediction accuracy of PMDs. [30, 40].

3.3.3. Intent Inference

A major drawback of the PMBs is the look-ahead time. PMBs essentially propagate the current and sometimes the past states into the future, a path prediction. Utilizing intent information can greatly increase the trajectory prediction along a route. Extracting the information on what the aircraft is most likely doing to do is referred to at intent inference. Intent inference does not only refer to waypoint-to-waypoint intent but also as a consequence of other information, such as weather patterns that might alter the planned route. [21] A visual explanation of what intent is is seen in Figure 3.8.

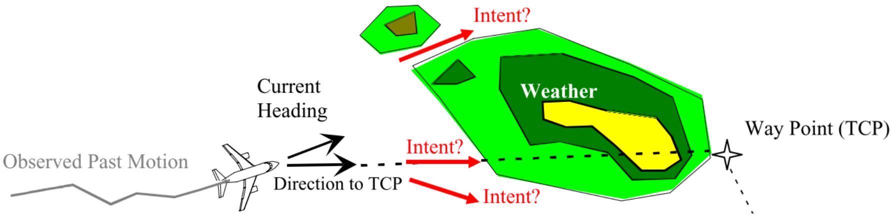


Figure 3.8: Understanding what aircraft intent is. Not only the waypoint determines the intent. [21]

Krozel et al. has made a match between the human flight control decision making process and intent inference. As Krozel et al. put it: "Intent inference is related to inferring the declarative and procedural decisions of the pilot, and path prediction is related to inferring the path that the pilot attains from regulatory and reflexive control inputs."[21]. This approach of looking at intent is a novelty and moves away from pure en-route intent prediction, making a gesture towards the possibility to derive aircraft intent in a FRA. For example, in Figure 3.8, the weather system can be replaced high a traffic complexity system, and the intent model still applies. Using sensory information that comes naturally in the human decision making process is a very important consideration to make when devising the air traffic complexity model. Since the information in this model should represent the same information on which a pilot makes meta-intent decisions. "Meta-intent" refers to the pilot intent at navigational level. Although this research is not concerned with PBMs, this way of thinking provides valuable insight into trajectory prediction in FRA and using environment information to generate a humanly intuitive air traffic complexity model. Figure 3.9 is a schematic control model that links the human decision making process to the aircraft control sequence.

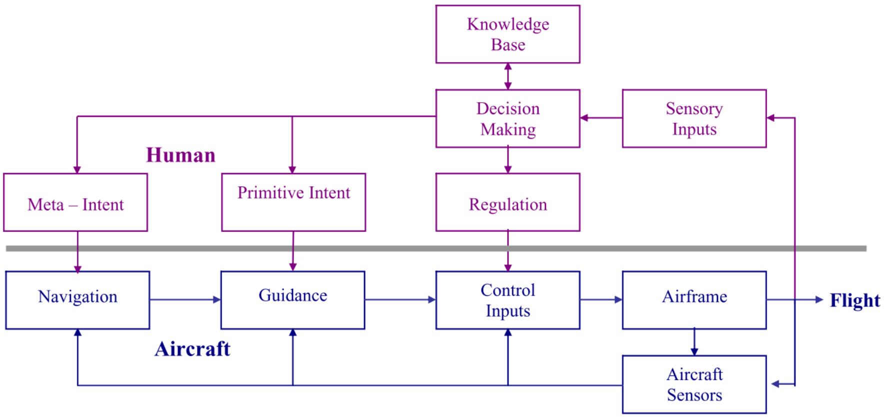


Figure 3.9: The human decision making process for flying an aircraft according to Krozel et al.[21]

Two more concepts taken from the intent inference research can be adapted to this research. First, a finite set of probabilistic intent models determine the range of possibilities, limiting the infinitely many elements that in fact determine intent. These models represent the range of possible pilot's action and are constrained by ATC regulations and procedures, such as 'hold pressure altitude of spatial location of waypoint'. [53] However, it is questionable whether this 'discretization' is necessary when applying a machine learning algorithm to determine intent. Lastly, the performance of online trajectory prediction and intent updating is improved by only updating the prediction when the deviations between actual and predicted exceed a certain predetermined threshold. This approach makes the prediction accuracy-driven instead being driven by a fixed look-ahead time, thus improving the reliability. [54]

3.3.4. Data-Driven Trajectory Prediction Models

In recent years, purely data-driven methods have been explored increasingly. These, naturally probabilistic, data-driven methods are independent from any aircraft model and sensory data, and have the ability to take into account various sources of uncertainty and learn from historical data. Although strictly speaking the application of a Monte-Carlo simulation on a PBM as mentioned in Section 3.3.2 is also data-driven, the notion of data-driven methods has a different meaning in this context. In this study, data-driven techniques allow trajectory predictions based on machine learning and agent-based modeling methods, considering all relevant, actual historical data, including contextual features. This means that these methods can encapsulate more parameters and underlying relations previously unforeseen with analytical methods. SESAR's *Data-driven AiRcraft Trajectory prediction research* (DART) project emphasizes the importance and potential benefits that research into this field will bring. This section will cover a few of the most relevant and distinguishable approaches to data-driven trajectory prediction.

Machine Learning: A General Overview

In machine learning we can classify algorithms based on the type of learning and based on the functionality, how they work. The three main learning styles are supervised learning, unsupervised learning, and reinforcement learning. In supervised learning the input data is called training data and has a known output. Generally speaking, supervised learning attempt to model a relationship between the target prediction output and the input features. The supervised learning class depend on the domain of prediction target: continuous variable or discrete (categorical) variable. A continuous function approximation is a regression problem. A discrete function approximation is a classification problem. In unsupervised learning, the training data does not include the desired output. In this case, the model has to mathematically organize the data according to unknown patterns or rules. Models that attempt to reduce feature redundancy are called Dimensionality Reduction algorithms. Models that attempt to group elements according so similarity are clustering algorithms. Lastly, Reinforcement Learning is a reward-based learning style in which a learning algorithm (agent) is trained by interacting with an environment. Reinforcement Learning will not be further considered after this point. Classifying algorithms based on the functionality allows one to understand the working principle, advantages, disadvantages and suitability to type of problem. An in-depth explanation of the algorithm will be done only on those that are applied in the related research. Many algorithms can actually be adapted for several types of learning problems. Neural networks, for example, can be applied to a regression, clustering, and classification problem. Figure 3.10 gives a illustrative overview of a variety of algorithms.[2]

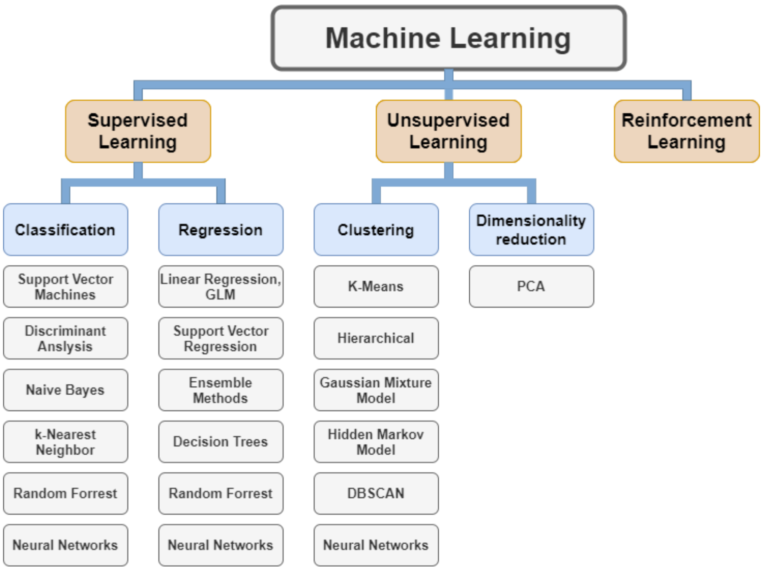


Figure 3.10: An overview of a variety of Machine Learning algorithms. Note that the algorithms are not all exclusive to each learning type.

Machine Learning Algorithms Applied to Trajectory Prediction

Machine learning is particularly suitable to the problem of trajectory prediction for several reasons. Firstly, there is more data becoming available every day. Second, it is a high dimensional environment. Many variables, including aircraft performance parameters, aircraft characteristics, and many external variables *may* play a role in the decision and execution of a trajectory, making it nearly impossible to find an analytical solution. Each external variable has a distinct distribution and influence on the trajectory, including: atmospheric conditions, convective weather, strategic routing, ATC interventions, conflicting air traffic, air traffic complexity, availability of airspace, airspace structure and even calendar properties. [15, 42] If any of these variables respond to a pattern, machine learning algorithms might identify them once the proper system features are considered. Lastly, the output variable are well defined and understood, spatial coordinates or *Estimated Time of Arrival* (ETA) at certain predefined points.(delay) This makes it intuitively comprehensible, thereby making the selection of algorithm type, hyperparameter tuning, and accuracy analysis feasible. The complexity of the problem can grow exponentially if the spatiotemporal boundary conditions are expanded (sector size and look-ahead time). The problem formulation can be a non-linear regression problem, where the trajectory represents a function that needs to be modelled. Alternatively, some researchers opt for a classification problem. In simple terms, this means that at every state, a discrete set of choices can be made on the propagation of that state. Lastly, clustering is also frequently observed, grouping sets of trajectories and making predictions based on the identified clusters. Frequently, data-drive trajectory prediction methods utilize a combination of techniques. This makes it rather cluttered to categorize each approach since there is much overlap. This section will cover the previous work grouped by similarly of the method, and not necessarily by algorithm or learning type.

- **Linear Regression Model**

  Published in 2013, Leege et al. [8] trains a *Generalized Linear Model* (GLM) to predict the time over points along the fixed arrival route. The GLM is a supervised regression problem. Since the prediction

[2] https://nl.mathworks.com/help/stats/machine-learning-in-matlab.html

is performed along a fixed route, the output variables are relatively low dimensional. A GLM is in fact not very different to a ordinary linear regression model, where a linear predictor is used to predict the output variables that have a certain probability distribution. A novelty is that Leege et al. applied a stepwise regression approach to determine the explanatory power of each input variable. The distribution of the output metrics are then compared for the varying input variable to understand the effect of each input variable on the model accuracy. Figure 3.11 shows this distribution. It can be seen well that at a 15NM prediction horizon, the addition of altitude and speed significant reduces the time error. This concept of introducing input variables with a stepwise approach to observe the model accuracy can be very valuable in understanding the explanatory power of the input variables. This is a common issue for 'black box' models. However, a GLM is expected to not capture the highly dimensional interaction effects present in this research problem.

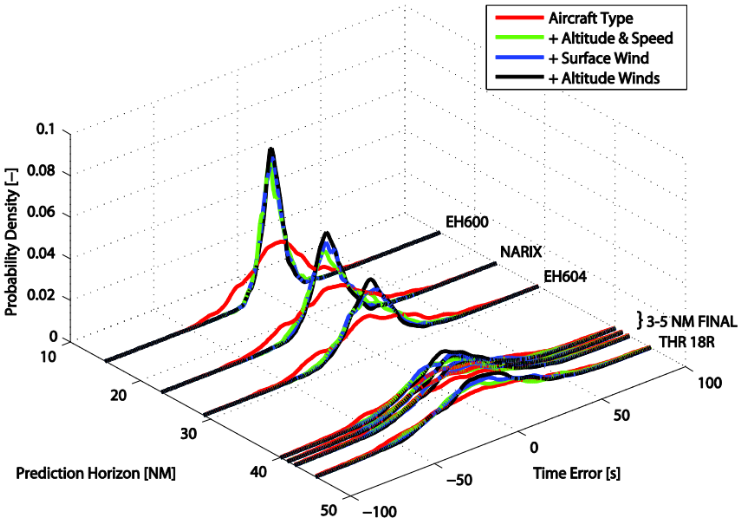


Figure 3.11: The Probability density distribution of the time error for GLM with varying inout variables. [8]

- **Markov Models**

  In 2006, Choi et al. was one of the first to represent a trajectory as a Markov Model [34]. This was a novelty since a Marvok model relies on a discrete system representation and is particularly suitable to stochastic systems. It means that a trajectory has to be broken up in segments, where the transition between states is represented by a stochastic model, which is learned from the data. Ayhan et al. further developed this idea and applies the *Hidden Markov Model* (HMM) to 4D aircraft trajectory prediction, taking into account atmospheric uncertainty [4]. Ayhan et al. represents the airspace as set of 3D cubes, where each cube contains homogeneous environment data. A trajectory is then represented by a sequence of these cubes with spatiotemporal attribute including weather conditions, as seen in Figure 3.12.

  A learning HMM is applied to represent the transitions between these segments and subsequent the Viterbi algorithm to compute the sequence of transitions, or the trajectory. A horizontal accuracy of 12.6 km is achieved (mean cross-track error). However, the highest spatial resolution was 13 km, so this proved to be the most limiting factor to the accuracy. If a similar approach is chosen in this research, which is possible due to the grid representation of the air traffic dynamics, careful consideration must be made to the grid size. The learning algorithm is very suitable to this problem representation. [12]

- **Neural Networks** The straightforward principle of a *Neural Network* (NN) does allow it to be adapted to a variety of tasks. NNs can be applied to classification, regression and clustering problems and have been used for trajectory prediction research as early as 1999. In more recent years, adaptations of NNs have been applied to the trajectory prediction problem. This literature review will focus only on recent developments. A much cited researcher, Daniel Delahaye, co-authored several research papers on ATM and 4D trajectory prediction. Together with Delahaye, Wang et al. [49, 50] performed a direct comparison between MLR and NNs and concluded that that NNs outperform MLR for the trajectory

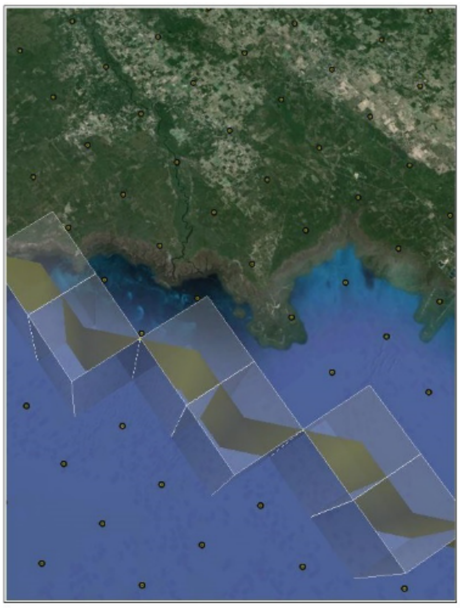


Figure 3.12: 'A set of spatio-temporal data cubes defining an aligned trajectory'[4] The dots represent weather observation nodes.

prediction task. This work was extended by comparing shallow NNs to deep NNs. The most straightforward adaptation of a single-layered (shallow) NN is by adding hidden layers, to generate a so called deep NN. It is shown that deep NNs consistently outperform shallow NN's . This is especially true for noisy data with a large number of outliers. However, excessive numbers of hidden layers (three in this case) causes overfitting, thus deteriorating accuracy and computational efficiency. [51] Next, *Recurrent Neural Networks*(RNNs) are especially suitable for sequence modeling and employing (time-varying) spatiotemporal patterns, as observed with (multiple) aircraft trajectories. *Long Short-Term Memory* (LSTM) is a popular adaptation of a RNN and has been proposed and researched for the application of trajectory prediction.[26, 55] The LSTM NN has a memory function because the output of the LSTM NN depends on the previous calculation results and the current input. For certain time-series prediction tasks, it has been shown that LSTM NN outperform the autoregressive moving average (ARIMA) model as well as the *Support Vector Regression* (SVR) model. [16, 25] Lastly, LSTM has been shown to also outperform Markov Models for flight trajectory Prediction by Shi et al. [44] Markov Models, as mentioned, are know to performed well for time-series processing. This result, along with the related work on LSTM and deep NNs, prove that adaptations of shallow NNs are suitable for trajectory prediction and are serious contenders to be applied in this research.

- **Ensemble Meta-Estimators** Lastly, ensemble machine learning methods, are known to be powerful estimations applied to trajectory prediction problems. A clear definition is given by Hernadez et al. [17]: "Ensemble-meta estimators can be defined as a collection of multiple learning algorithms that average the predictions from multiple models to yield a final prediction." This method can be applied to both regression and classification problems, but is most suited to classification problems. Applied ensemble methods are *Gradient Boosting Machines* (GBM) and random forest models, which have been shown to be very suitable to the trajectory prediction problem.[9, 17] In direct comparison, GBM and random forest models perform similarly and outperform other ensemble techniques as well as logistic regression. [6, 27] Note that these last two studies applied a classification problem. In the only study found to compare LSTM to GBM for trajectory prediction as a regression problem, Dek et al. [9] concludes that GBM is the superior machine learning algorithm. However, this study does apply clustering and is applied in an environment with highly repetitive flight trajectories and short look-ahead times. This problem favors GBM because of the categorical indicators and high similarity between flight situations, making it almost resemble a classification problem.

Hybrid Clustering and Trajectory Prediction Methods

A commonly observed approach to trajectory prediction is that of a hybrid clustering and machine learning method. This method is widely applied in the context of trajectory prediction since it greatly reduces the computational complexity and for its explanatory power as shown by Wang et al. [50], Marcos et al. [31], Liu et al. [28], Lui and Klein et al. [29] and Dek [9]. The most common methods for clustering are K-means clustering and *Density-Based Spatial Clustering of Application with Noise* (DBSCAN). Generalizing, DBSCAN performs well with large data sets and in the detection of outliers. Clustering can help identify the feature importance and often works well with supervised learning algorithms for classification problems. Note that clustering itself is not a prediction method, it has to be combined with a prediction algorithm that considers the features on which the clustering is based and places a trajectory in a cluster. There are two general applications of clustering in the context of trajectory prediction.

First is trajectory clustering and nominal route prediction. The clustering algorithm is used to identify route choices based of relevant features. The machine learning algorithm is used to model the aircraft route choice, placing a trajectory in a cluster. The actual trajectory prediction is then the (usually) median of that cluster, see Figure 3.13. This essentially makes it a classification problem. The machine learning algorithms that have been applied are Logistic Regression, SVR, Random Forest, GMB, and HMM. [15, 28, 29]. As Lui et al. mentions: "...instead of directly predicting individual flight tracks, [the approach] predicts the aircraft route choices by consolidating a large number of trajectories into several clusters within which flight tracks are similar to each other."[28] The use of clustering and nominal trajectory prediction should be carefully scrutinized as it can potentially contradict the objectives of this particular research, as it relies or recurrent trajectories and standard routes.

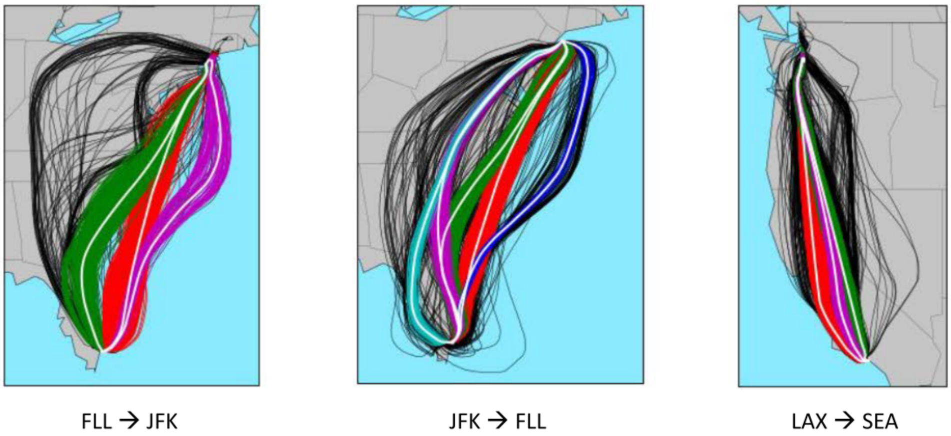


Figure 3.13: An example of three routes, each with a set of clusters. The white line represents the median 'nominal' route. [28]

The second hybrid approach consists of clustering to assign a trajectory to a cluster and subsequently followed by a machine learning algorithm to predict the individual trajectories within that cluster. This has been done mostly for aircraft inside the TMA that follow the standard arrival/departure procedure and regular ATC instructions. [50**?**] The machine learning algorithms applied range from Multi-cell NN, Multiple Linear Regression Model, Muli-layer Perceptron, GBM, and LSTM NNs. In is important to note that this hybrid approach only uses clustering to generate a subset with similar trajectories, as a dimensionality reduction. The trajectory prediction step in fact is no different to the pure machine learning approaches. Therefore, these studies will be taken into account to compare suitability, accuracy and efficiency for this study.

Summarizing, it is not expected that clustering will be needed for this research for three reasons. Firstly, this research is concerned with FRA and en-route data, aiming to make trajectory predictions that are independent of standard 'nominal' routes. Secondly, generating a subset to reduce the dimension is not expected in this research due to the availability of aircraft intent information. Aircraft intent is expected to already narrow down the range of possible trajectories enough. Third, the en-route data does not visually appear to have major reoccurring routes that can be clustered well, see Figure 3.14. However, if the model complexity and highly-dimensional problem prove to be too much for the application of this research, clustering is a tried-and-tested approach to reduce the dimensionality.

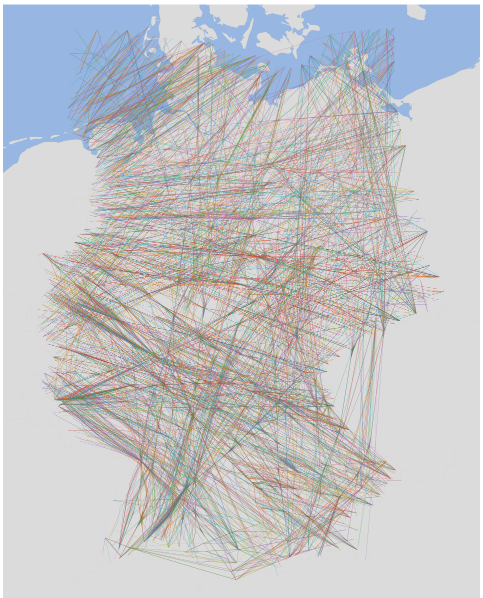


Figure 3.14: All the trajectories of 01-09-2018 in the selected region.

3.3.5. Trajectory Prediction Metrics

The scope of the trajectory prediction can range from single delay prediction upon entering the airspace to 4D trajectory prediction with online intent updating and a long look-ahead time. The required output, evaluated by certain metrics, dictates what algorithm is most suitable. The evaluation metrics are based on the difference between the predicted an the actual trajectory. The metrics that are observed are :

- ETA or delay prediction error. This entails a fixed location and a estimated time.
- Spatial prediction error. This entails a fixed time and a estimated position. This could be along any, or multiple axis. Cross-track error, along-track error, horizontal-track error and altitude error are among the observed metrics. See Figure 3.15 for a geometrical representation of the (horizontal) spatial errors.
- Flight properties. Other properties can also be included in the prediction, or may be used to determine the time or spatial error. Such properties include speed and heading angle.

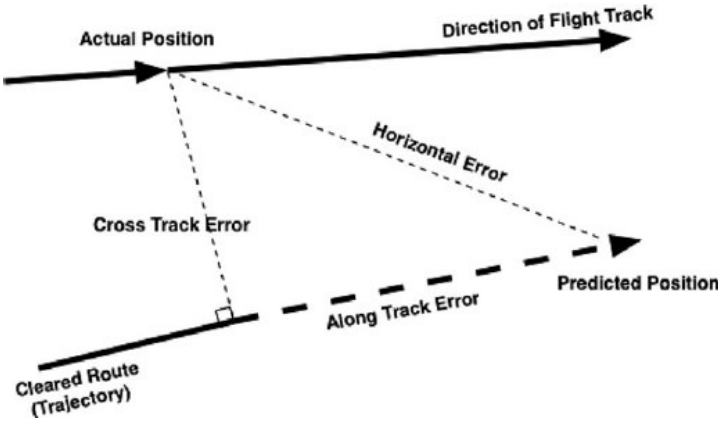


Figure 3.15: Geometric overview of horizontal, along-track, and cross-track error. [39]

The navigation accuracy is analyzed by X-percentile method, Circular Error Probable, *Root Mean Square Error* (RMSE), and x-sigma. It is chosen that the RMSE and/or MSE is most suitable. In case of a classifier, a ROC curve is used to evaluate the accuracy. [6] The errors are evaluated based on a number of variables. A critical variable that is very relevant for this research is the navigation accuracy at various look-ahead times.

3.4. Input Data

3.4.1. Eurocontrol RD dataset

The data source used in the first phase of the research, understanding the relation between air traffic dynamics and aircraft trajectory, is from the Eurocontrol RD data archive. This data is collected from all commercial flights operated in and over Europe, including flights that depart outside of Europe but arrive inside Europe, and vice versa. It is a processed data set to ensure accuracy that includes data from air navigation service providers' flight data systems, radar, and data link communications. According to Eurocontrol, this results in data that "represents the best view as used by air traffic management" [13].

The data is historic data of all commercial flights (excluding military, state, and general aviation flights) from four sample months: March, June, September, December. The data includes flights from the years 2015, 2016, 2017, and 2018. This is large-scale data, including access to seasonal patterns suitable to the research objective. The data consists of the following metadata that is used in this research:

- **Flights** A list of flights with key flight details. Below are two sample rows including the data fields. The data from September 2018 returns just under 1 million flights.

ECTRL ID	ADEP	ADEP Latitude	ADEP Longitude	ADES	ADES Latitude	ADES Longitude	FILED OFF BLOCK TIME	FILED ARRIVAL TIME	ACTUAL OFF BLOCK TIME	ACTUAL ARRIVAL TIME	AC Type	AC Operator	AC Registration	ICAO Flight Type	STATFOR Market Segment	Requested FL	Actual Dist. Flown (nm)
222570350	EPKT	50.47417	19.08000	LGTS	40.51972	22.97083	01-09-2018 00:00:00	01-09-2018 01:49:37	31-08-2018 23:56:00	01-09-2018 01:45:42	A320	ZZZ	LZMDK	N	Charter	370.0	647
222570351	KOAK	37.71667	-122.21667	ESSA	59.65194	17.91861	01-09-2018 00:00:00	01-09-2018 09:27:15	01-09-2018 00:06:00	01-09-2018 09:42:52	B789	NAX	LNLNL	S	Lowcost	370.0	4880

Table 3.2: Sample data from 'Flights' dataset

- **Flight points** Both actual and filed flight points of all flights as in 'Flights' metadata. The data from September 2018 returns in the order of 30 million filed and 30 million actual flight points. Below are two sample rows including the data fields.

	ECTRL ID	Sequence Number	Time Over	Flight Level	Latitude	Longitude
0	222570356	76	01-09-2018 08:58:30	400	50.19056	12.33861
1	222570356	77	01-09-2018 09:03:05	400	50.37861	11.42528

Table 3.3: Flightpoint example data entries

- **Flight Through Airspaces** Both the filed and actual entry and exit time into a *Flight Information Region* (FIR) and *ATC Unit Airspace* (AUA) for each aircraft. The dataformat of FIR and AUA metadata is identical, apart from the FIR/AUA ID. Below is a sample row including the data fields.

	ECTRL ID	Sequence Number	FIR ID	Entry Time	Exit Time
0	222570350	0	TAXI_OUT	31-08-2018 23:56:00	01-09-2018 00:06:00
1	222570350	1	EPWWFIR	01-09-2018 00:06:00	01-09-2018 00:16:26

Table 3.4: FIR entry and exit example data

- **ATM environment data** Three sets of metadata that refer to the structure of the airspace are also provided. It is not expected that this data will be used.
  - **AIRAC** This dataset defines a series of common dates and an associated standard aeronautical information publication. Not used for this research.
  - **Routes** Includes route identifier, route point sequence number, and the coordinates of that point on the route. Not used for this research.

– **FIRs** Includes the coordinates and flight level range of the boundary of the sector, both FIRs and UIRs.

3.4.2. Flight Region and Date Selection

The entire geographic extent of the available data is not applied. This has several reasons.

- **Free Route Airspace** As explained in Section 3.1, in line with the objective of this study a sector of airspace should be studied where an aircraft has some degree of autonomy over the chosen route on a tactical level. Since the most recent available data is from 2018, it is preferable to apply this study to the dark green airspace sectors in Figure 3.3.

- **High Congestion** In the first phase of this study it is preferred to study a frequently congested airspace to capture any air traffic dynamical effects on the flown trajectory. If the airspace is barely congested, is it expected an aircraft does not have to deviate from its planned trajectory as frequently. So, for the statistical analysis higher levels of congestion are expected to reveal more patterns. In the second phase of this study however, it is important to include low levels of congestion in the training phase in order to understand how the different levels of congestion lead to different trajectory changes and secondly, be able to create robust predictors. The September 2018 Monthly Network Operations Report reveals that from the available months and years of the Eurocontrol RD dataset, September 2018 had the highest ever recorded daily traffic levels at around 34,000 flights. [13] This corresponds to the cumulative number of flights of around 1 million as seen in the Eurocontrol RD dataset. It is chosen to use data from this month for the first phase of this study.

  The region that experienced the highest level of en-route delay in September 2018 is the Karlsruhe *Upper Area Control* (UAC), as can be seen in fig. 3.16. This does not mean that the flights got delayed inside the respective UACs. On the contrary, it is expected that much of the delay occurs prior to entering the sector since the capacity is limited and so flights have to wait before entering a sector. This often occurs prior to take-off.

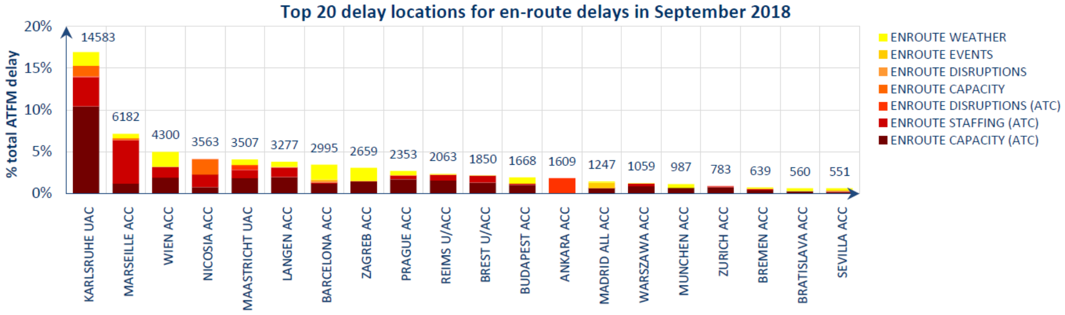


Figure 3.16: The distribution of delay per region in Europe.[13]

- **Airspace size** Since the analysis and trajectory prediction will be performed on en-route flights, the sector size must be large enough for cruising aircraft to spend significant time in the sector. This is needed in order to capture enough trends and also be able to make med-long term prediction. It is possible to include data from many adjacent sectors. However, it is expected that differences in controller behavior and/or policy in different areas lead to different air traffic dynamic behavior. It would then not be possible to capture this behavior well in a single model.

The above mentioned considerations for selecting the appropriate time and sector for this study has led to the airspace under control of Karlsruhe UAC and part of Maastricht UAC during Sepember 2018 to be selected for the first statistical phase of the study. These are the FIRs of Upper Airspace Hannover UIR (EDVV) and Rhein UIR (EDUU), both at flight level 245 and above. If during the analysis this proves to be too small, the area of consideration will be extended with the remaining Maastricht UAC UIR's: Amsterdam UIR (EHAA) and Brussels UIR (EBUR). For the trajectory prediction phase, careful consideration must be taken to select representative data that has enough datapoints and variability.

3.5. Summarizing Remark on the Literature Survey

The goal of the literature review is to provide an academic framework on which this thesis can build. The two dominant topics in this literature review are the air traffic dynamics and the trajectory prediction. The literature review provides a starting point on these topics. Derived from the literature review are the proposed air traffic dynamics model and the method for trajectory prediction. Although these are direct results of the literature study, the proposed models and algorithms are summarized in Section 4.1 and Section 4.2 as part of the research plan.

4 Trajectory Prediction Research Plan

This chapter will give an overview of the experimental phase to be conducted in this thesis, the trajectory prediction. The research objective is to improve the performance of the medium to long term trajectory prediction by incorporating the air traffic dynamics. The work in the first phase of this research is focused on quantitatively representing the air traffic dynamics and selecting which model, or features of the model, that are expected to be most suitable as part of a trajectory predictor. This chapter will first include the proposed models, both for the air traffic dynamics in Section 4.1 and for the trajectory prediction in Section 4.2. This is derived and based upon the literature review in Chapter 3.

Next, this chapter will discuss how the data is prepared for the trajectory prediction using the LSTM NN. This includes the defining of the input variables, the output variables, and steps that will be taken to reduce the dimensions. If during the execution of the experimental phase of this study it is concluded that LSTM NNs are not the best algorithm, parts of this chapter might become obsolete. Next, the trajectory prediction experiment will be introduced. Since this experiment has not been conducted yet, this section will give insight into the baseline experiment, the steps taken to perform the trajectory prediction, as well as means to perform the validation of the trajectory prediction.

Note that the research plan for conducting the preliminary statistical analysis is not covered in this section. This is because this phase has been conducted and the preliminary results are discussed in Chapter 5.

4.1. Proposed Air Traffic Dynamics Models

Following the literature review, it is selected that the following three air traffic dynamical models are most suitable for the application of this research. Although ingenious, most advanced complexity models don't lend itself well to the context and application of this research. The argumentation of each model and if necessary specification is given below. Essentially it comes down to three different representations: one captures the most simple and basic features which might be considered by ATC and pilot during tactical trajectory prediction; The Lyapunov exponent is an aggregated mathematical measure of disorder; Lastly, the Eurocontrol Complexity Metrics represent the one factor which ATC and pilots attempt to avoid, which is the number of interactions between aircraft.

Regression Model

The proposed features that will constitute the regression model are based on the Dynamic Density model, the TBO complexity indicators, and one feature from the Dynamic Weighted Network. This selection is based on the literature study, evaluating each feature based on the following: Controller independence, Route independence, objectivity, and expected complexity indicator. The features that have been chosen have widely shown the highest correlation with (subjective) complexity ratings from Radisic et al. [38], Andrasi et al. [3], Szamel et al [46], and Wang et al. [48]

Traffic Density At various scales depending on sector area (TD);

Aircraft count At various scales depending on sector area (ACC);

Heading change Count of aircraft making >15 degree heading change within 2 minute period (HdgCnt) and average heading change for aircraft passing this threshold (HdgAvg);

Speed change Fraction of aircraft with an airspeed change of >10 kts within a 2 minute period (SpdCnt) and average airspeed change for aircraft passing this threshold (SpdAvg);

Altitude change Fraction of aircraft making >750 ft altitude change within 2 minute period (AltCnt) and average heading change for aircraft passing this threshold (AltAvg);

Minimum distance Count of aircraft pairs at 3D Euclidean distance less than 5 NM (MinDst5), 5-10 NM (MinDst10), and 10-50 NM (MinDst50) separation. Possibly be split in vertical and horizontal separation;

Average Distance Average weighted horizontal distance between all aircraft(AvgDst);

Heading Vaiance Standard deviation of aircraft headings (SDHdg);

Conflict predicted Conflict predicted. Fraction of aircraft predicted to be in conflict within 600 seconds (Cfl);

Convergence Rates Between-aircraft complexity, based on spatial approaching rates, see eq. (3.2) (ConvRt).

Lyapunov Exponent

In contrast to separate, basic features, that each contribute to complexity with various weighting, the Lyapunov exponent provides a method to compute the complexity in a single indicator. It is a intrinsic measure of complexity measuring level of order, convergence, and sensitivity to initial conditions. A risk however, is that the Lyapunov Exponent complexity measure does not measure the same entity that is determinant for trajectories. This will have to be proven during the feature selection and engineering phase. There exists a possibility that there is cross-correlation with convergence rates as named above since it is a element of the Lyapunov Exponent.

Eurocontrol's Complexity Metrics

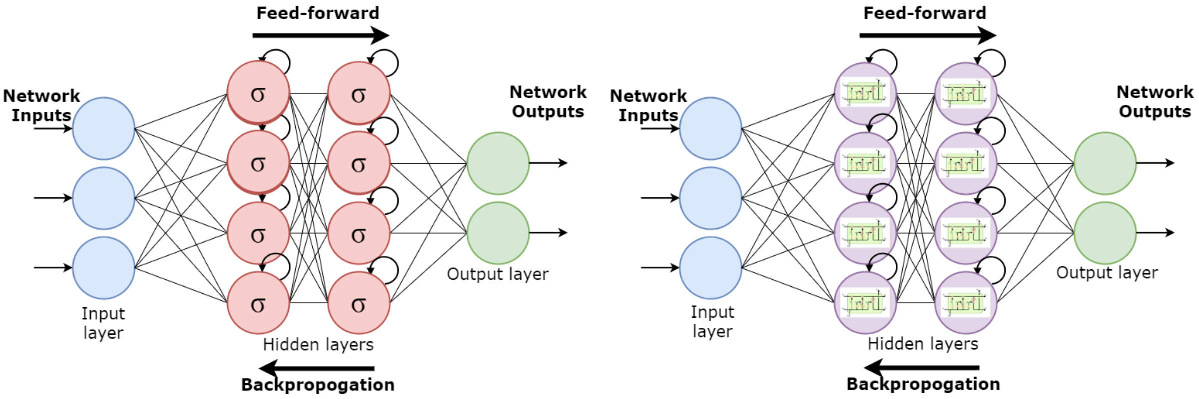
The complexity metrics proposed in Eurocontrol's study provide a simplified complexity model intended to express the air traffic dynamics as basic and intuitively as possible. This way, the air traffic complexity is expressed similarly to the how the decision making unit would process the information. This is a novel and unique method that is very different to the other models and therefore worthy to consider. It's results are expected to be different to the previously chosen air traffic complexity indicators. Also it is well adapted to 'add' to the Dynamic Density features. There is a possibility that this measure correlates with the *Heading, Speed,* and *Altitude change* features from the regression model .

4.2. Proposed Trajectory Prediction Algorithm

In line with the objective of this research, a data-driven trajectory prediction algorithm is chosen instead of a PBM. It is chosen to not initially apply clustering for dimensionality reduction before the training of the trajectory predictor for the following reasons: The en-route flight profiles in upper airspace are less dependent on standard routes than, for example, TMAs, especially in regions with FRA capability. It is often seen that hybrid prediction methods are performed on either standard routes or within TMAs.; The selection of route-independent air traffic dynamical models make it likely that the features will not correlate with any route information.; Lastly, *Principle Component Analysis* (PCA) is a well-known and effective method to reduce the dimensions and feature redundancy prior to the training of any supervised machine learning algorithm. The most promising machine learning algorithms are HMM, GBM, and LSTM. These methods have proven to be suitable for trajectory prediction, albeit under different conditions. A LSTM NN is the preferred choice of trajectory prediction algorithm because of the following reasons: The absence of routes, SIDs, or STARS and clusters make it unlikely that the supervised learning problem will resemble a classification problem, making a algorithm optimized for regression a better choice; The 4D trajectory prediction is expected to depend not only on the most recent datapoint, but on a sequence of datapoints and delayed effects of air traffic dynamics, making a LSTM the method of choice. Note that without performing tests and perhaps a direct comparison it cannot be said with certainty which methods performs the best, especially when taking into account computational effort as well (LSTM are quite heavy relative to GBM and HMM). If during the execution of the final phase the results are not acceptable, a re-evaluation will need to happen.

The Long Short-Term Memory Neural Network

To understand the architecture and basic working principle of a LSTM, a description will be given. It is assumed that the fundamental working principle of a NN is understood. A RNN is different from a Feed Forward NN since it has a recurrent condition on each module of the hidden layers, which ensures that sequential information is captured in the input data of each hidden sate: internal memory. The difference between a RNN and a LSTM NN is the way in which the memory is stored, see Figure 4.1 for the general structure of a RNN and LSTM.



(a) Recurrent Neural Network

(b) LSTM Neural Network

Figure 4.1: Architectural differences between RNN and LSTM NNs

The RNN suffers from long-term dependencies when the 'distance' between the relevant information x_t and place it is needed h_{t+i} become too large. The weights of the NN are updated proportionally through back-propagation and become very small after several time steps, until the network practically stops learning: the Vanishing Gradient Problem. The LSTM is designed to deal with these long-term dependencies. In Figure 4.2, the LSTM network is visualized. Where **A** represents the recurrent module of NN, x the input vector, and h the output. RNNs have a single recurrent NN module, of which the activation function is often *tanh*, this can be seen to be different for the LSTM.

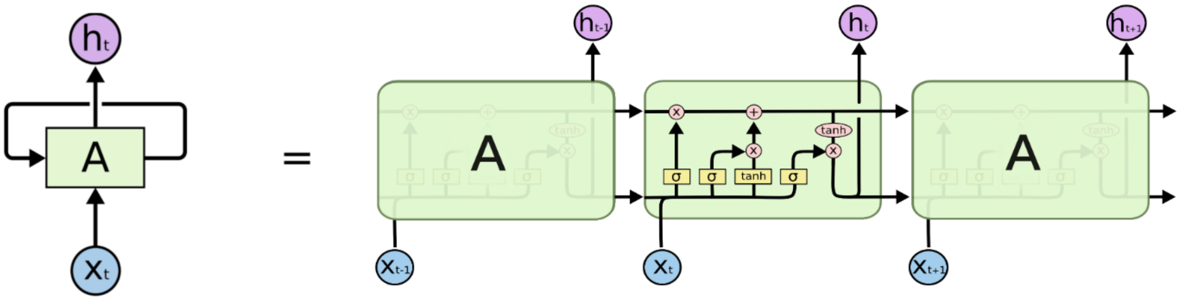


Figure 4.2: A unrolled RNN with the repeated module showing the LSTM with four interacting NN layers. [33]

The horizontal line across the top is the cell state, which carries the information through the entire chain. The LSTM module adds and removes information to the cell state through structures called gates. This way the cell state only relies on relevant information. There are three gates: the forget, the input, and the output gate. Each gate contains a sigmoid (σ) layer which passes through either all the information (0) or none (1). The input gate also contains a $tanh$ layer ([-1,1]), hence the four interacting layers.

The forget gate consists of a single sigmoid layer and can store or disregard the information from the cell state at $t-1$: keep or forget current cell state. The input gate controls the information that will be added to the cell state. The input gate first decides to update the cell state or not through the sigmoid layer. The $tanh$ layer creates new feature values which is used to update the cell state. The output gate 'filters' the cell state and determines how much of the updated cell state to pass through as an output. First the sigmoid layer decides which part to output, which is then passed through a $tanh$ function to map it between -1 and 1.

4.3. Data Preparation

This section will outline the data format required for LSTM NNs including a few processing steps, means for dimensionality reduction, and the input variables, output variables.

4.3.1. LSTM NN data format

The famous machine learning credo of 'Garbage in, garbage out' is as relevant for NNs as it is for all other Machine Learning algorithms. However, not only does the input data need to be well understood and structured, it must also adhere strict coding formats. In Python, LSTM NNs can be implemented with the Keras[1] and Tensorflow[2] library. The LSTM NN input variables requires a 3D format of `[samples, time steps, features]`. The samples are each flight, consisting of n timesteps, which correspond to the datapoints. If the amount of datapoints are not sufficient, then the flightpoints will be interpolated with shorter timesteps. So the shape of the input data will be: (num of flights, num of flightpoints, num of features).

Some features that will be implemented are categorical text features. These can be converted to numerical variables by using dummy variables. This is done by allocating a column to each categorical element and setting a column to 1 if the sample falls in the category and 0 otherwise: binary vectoring. For example, say there are three aircraft type in a training data-set: A320, B787, and A350. As a variable, these would be formatted as: [1,0,0], [0,1,0], and [0,0,1], respectively. Applying dummy variables as such is also referred to as One-Hot-Encoding, but the wording dummy variable will be used in this report.

Training a network on unscaled data can possibly slow down the learning and converging of a NN. A variable with mean value of 1000 will initially be much more dominant over a variable with a mean value of 0.1. This can introduce 'bias' into the network, pushing the network towards local optimum, slowing convergence and in extreme cases it can even prevent the network from converging at all. This is a very common issue, also for analytical regression methods. Feature scaling can mitigate this issue. Standardization, also called z-score normalization, rescales the distribution of samples around a mean of 0 and standard deviation of 1. Min-Max normalization rescales the samples between 0 and 1. Standardization is most suitable for variables with a Gaussian sample distribution. As is observed in Section 5.3, this is not the case for most variables. Therefore min-max normalization is deemed to be the better rescaling approach, and is done according to Equation (4.1).

$$v' = \frac{v - v_{min}}{v_{max} - v_{min}} \tag{4.1}$$

4.3.2. Geographical Coordinate System Transformation

The input data is defined in a spherical coordinate system (latitude ϕ, longitude λ, height h). For the LSTM, we must transform the coordinates into 3D Cartesian coordinates (X_c,Y_c,Z_c), defined with respect to the local tangent reference frame with East, North, and Up (ENU) conventions. This is done in two steps, first transform the spherical coordinate system to Cartesian coordinates w.r.t. the Earth-Centered, Earth-Fixed (ECEF) reference frame and then convert this to the local tangent reference frame. Figure 4.3 depicts the three reference frames.[3]

The spherical coordinates are converter to Cartesian coordinates w.r.t ECEF reference frame by the following Equation (4.2).

[1] https://keras.io/api/layers/recurrent_layers/lstm/

[2] https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM

[3] https://en.wikipedia.org/wiki/File:EarthTangentialPlane.png

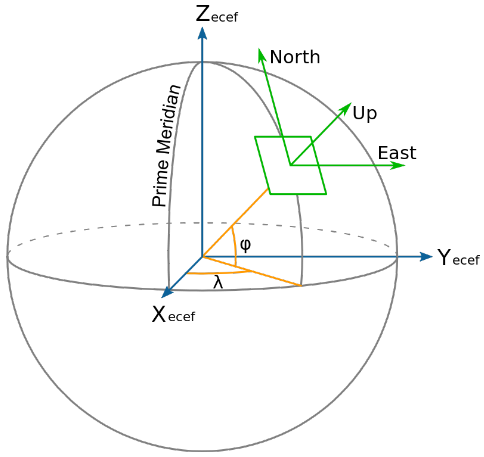


Figure 4.3: The three reference frames, spherical (yellow), ECEF (blue), and ENU (green)

$$\begin{aligned} X_c &= N(\phi) + h\cos\phi\cos\lambda \\ Y_c &= N(\phi) + h\cos\phi\sin\lambda \\ Z_c &= N(\phi) + h\sin\phi \end{aligned} \tag{4.2}$$

Where,

$$N(\phi) = \frac{a^2}{\sqrt{a^2\cos^2\phi + b^2\sin^2\phi}} \tag{4.3}$$

Where a and b denotes the equatorial and polar radius. To convert to ENU reference frame. The reference point, $[X_r, Y_r, Z_r]$ is taken at lowest latitude and longitude of grid representation. The data points are denoted $[X_g, Y_g, Z_g]$.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} -\sin\lambda_r & \cos\lambda_r & 0 \\ -\sin\phi_r\cos\lambda_r & -\sin\phi_r\sin\lambda_r & \cos\phi_r \\ \cos\phi_r\cos\lambda_r & \cos\phi_r\sin\lambda_r & \sin\phi_r \end{bmatrix} \begin{bmatrix} X_g - X_r \\ Y_g - Y_r \\ Z_g - Z_r \end{bmatrix} \tag{4.4}$$

4.3.3. Input Variables

Flight Characteristics

It is expected that airline operators, aircraft models, departure and destination airports, and type of flight play a part in the tactical trajectory planning and execution. To illustrate, each FIR charges an aircraft for passing through. Low-cost flights are expected to avoid 'expensive' FIRs, whereas higher market segment flights might opt for direct routes, saving valuable flight time. For this reason, it is valuable routing information that will be included in the the network model. None of the following features will be included as a time-series varying feature. They are stationary features for an entire trajectory (sample). Table 4.1 summarizes the aircraft characteristics input features.

Feature	Unit	Feature Engineering
Departure Airport	-	dummy variables
Destiation Airport	-	dummy variables
Aircraft Type	-	dummy variables
Aircraft Operator	-	dummy variables
ICAO Flight Type	-	dummy variables
STATFOR Market Segment	-	dummy variables

Table 4.1: Flight characteristics that will be used as features in the NN.

Flight Points

Naturally, current flight position data is crucial to predict the trajectory of a flight. Only the current position will be included for each sample. This is possible since the LSTM model is designed for time series and utilizes the data from the past. If the timesteps are all equal, the time does not actually need to be specified as a feature. However, since the time itself might influence controller strategy and procedures, it is used as a feature. At night the flown routes differ from daytime. Time is taken as seconds from midnight. This means that the day of the week will be a feature too, so that weekend effects are considered. It is not expected that the day of the month will play a role in the flown trajectory. If the computuational time allows, the NN can be trained on multi-year and multi-month data. The month will also be regarded as a feature to account for seasonal effects. However, to be conservative, this will be included initially. The architecture of LSTM NNs allows one to include a sequence of flight points. If this proves successful, then a time series sequence of past flight points can be included as an input Table 4.2 summarizes the flight point input variables.

Feature	Unit	Time-series	Feature Engineering Required
Time of day	seconds	yes	Min-Max normalisation
Day of week	-	no	dummy variables
Month of year	-	no	dummy variables
Flight Level	feet	yes	Min-Max normalisation
Latitude (y)	-	yes	Min-Max normalisation Transfer to Cartesian Coordinate system (x,y)
Longitude (x)	-	yes	Min-Max normalisation Transfer to Cartesian Coordinate system (x,y)

Table 4.2: Flight points input variables

Aircraft Intent

The filed flight points are perhaps the best indicator for aircraft intent. It is reasonable to include as a feature since ATC controllers currently have access to the filed flight plan and with the arrival of TBO and next iteration of ADS-B, it is expected that aircraft will have access as well. Instead of taking the same sequence number of filed flight point, the nearest filed flight point ahead of the actual flight point will be taken into account. It is assumed that this is where the aircraft will be headed. The architecure of LSTM NNs allows one to include a sequence to flight points. If this proves sucessful, then the entire sequence of future filed flight points can be included as an input. The filed flight points are in the same format as the actual flight points, as given in Table 4.2. The only difference is that the upcoming filed flight points will be taken instead of the current and past. Moreover, month of the year and day of the week will not be included.

Air Traffic Complexity Features

The inclusion of these features into the LSTM is what this research is focused on. This step must be done with due diligence. The LSTM NN can take multivariate inputs, but each feature has a single input value for each sample and timestep. This means that it is not possible to include the entire 2D grid representation as an input to each sample at each time step. In fact, a similar approach as with the statistical analysis must be taken, where each flight point corresponds to a single feature value. This approach is inspired by Ayhan et al. [4] as summarized in Section 3.3.4 and seen in Figure 3.12. The grid size once again is a crucial parameter that must be carefully tuned, similar to the hyperparameters of any machine learning model. Since the grid feature values are a time-series, the past values can be taken into consideration. The features, selected from the statisical anaylsis are summarized in Table 5.3. Note that not all the features from the air traffic dynamical models that have been selected as part of the literature study have been computed and analyzed yet. This means that the exact features have yet to be determined. This includes conflict predicted, convergence rates, Lyapounov Exponent, and the Eurocontrol Complexity Metrics. Each feature will undergo Min-Max normalisation.

Moreover, it is chosen to also consider grid values just ahead of the current grid in the direction of travel, like a radial extension. This will be vital in the 'steering' of an aircraft in a certain direction, similar to the changing aircraft intent in Figure 3.8. A visual representation of this is seen in Figure 4.4. The size of this surrounding grid will have to be determined.

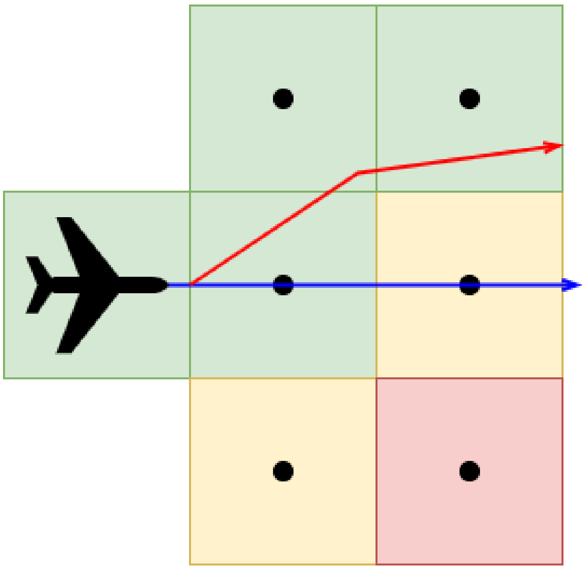


Figure 4.4: The grids 'in front' of an aircraft will also be considered to take into account surrounding complexity, but not have to consider the entire grid each time step.

4.3.4. Output Variables

Naturally, the output variables need to be carefully selected and defined. A trajectory is a sequence of spatio-temporal positions, expressed as numerical output. Since the input variables used are Cartesian coordinates, so will the output features that define the predicted trajectory. Altitude will also be included. The trajectory will be predicted as a time-series, this has many advantages. The architecture of the LSTM allows one to make predictions at incremental time steps. This means that a single network can make predictions at various look-ahead times.

Output variable	**Unit**	**Time-series**
Flight Level	ft	yes
Longitude	-	yes
Latitude	-	yes
Delay	seconds	no

Table 4.3: Output variables

4.3.5. Dimensionality Reduction

Dimensionality Reduction has already been conducted in the statistical analysis by feature selection. Further dimensionality reduction is conducted by actively performing a transformation on the data from a large set of variables to a smaller one. This will be done by *Principal Component Analysis* (PCA) . PCA transfers the variance of two variables and compresses redundant information. The new variables are the principal components and are uncorrelated. The PCA tried to place as much information in each variable as possible. By ranking the eigenvectors in order of the eigenvalue, the principal components are ordered. Disregarding the components with little variance will reduce the dimensions without losing much information.

4.4. Trajectory Prediction Experiment

In order to conduct the core experiment in this research, several steps need to be taken. It will be assumed that the air traffic complexity features have been calculated and all the input data formatted as presented in the previous section. A step-wise approach will be taken in which input features will be trained one-by-one. This is a necessary approach in order to satisfy the research objective and answer research question 5b. It will help make the trajectory prediction explainable, quantify the effects of air traffic dynamics features and move towards understanding the overall dynamics of air traffic. During every iteration, the performance of the trajectory prediction will be assessed based on the horizontal, cross-track, along-track, and vertical error for the positional output variables. The delay will be assessed based on a time-difference. The metrics have been discussed in Section 3.3.5.

4.4.1. Baseline Experiment

The baseline experiment is the first iteration of the the step-wise training of the NN. It will be the model that does not include any air traffic dynamic features. As the name suggests, this experiment provides a baseline to which all the other experiments are compared. Any performance differences compared to the baseline will be attributed to the air traffic dynamics features and carefully analyzed.

4.4.2. Cross-Validation

Prior to training the NN, we must split the dataset into three subsets: training data, validation data, and testing data. This is done to prevent overfitting. Testing data is altogether separated from the other subsets, and used to test the performance of the neural network on a fresh, unseen set of data. The general 80-20 rule is applied here, where 20% of the initial dataset is kept separated for testing. The remaining 80 % will be used to train and validate the neural network. The training data is used to actually fit the model. The validation data is used to frequently evaluate the model to tune the hyperparameters. Now the model does not actually use this model to fit the datas. However, since the data has an effect on the parameters and learning process, the model can become biased towards this validation dataset. Hence the use of a unseen testing dataset.

The choice of training and validation subsets can still coincidentally be biased. It is therefore recommended to reshuffle the training data and iteratively train and validate the model on randomized sets. The results of the iterative model runs are then averaged. This is called k-fold cross-validation and will be applied in this research. Note that the testing dataset is not involved, it is kept separated from the very beginning. A visual representation of this method is seen in Figure 4.5[4].

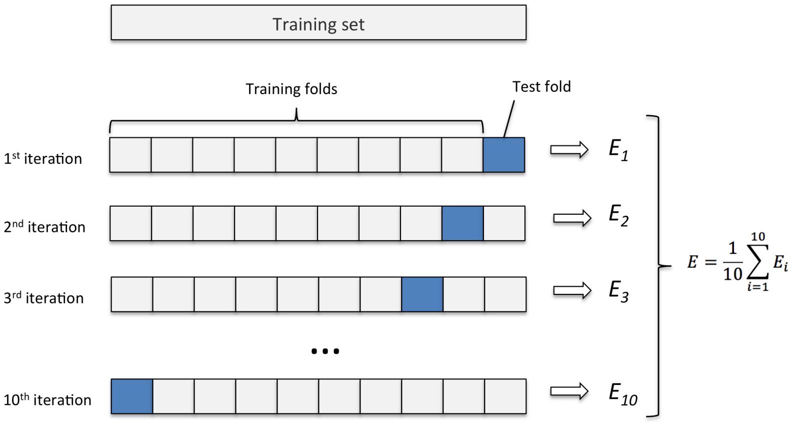


Figure 4.5: An example of 10 iterations of k-fold cross-correlation on the training dataset.

[4] http://karlrosaen.com/ml/learning-log/2016-06-20/

4.5. Evaluation, Validation, and Limitations

Evaluation

Evaluation of intermediate results is done primarily during the statistical analysis of the air traffic dynamics feature. By performing a carefully conducted statistical analysis, it is verified that there are relationships between the air traffic dynamics features and trajectory anomalies, which is indicative that it provides valuable information for the trajectory prediction. Throughout this phase, unit tests, integration testing, and frequent visual representation of the data ensure verification.

Verification of a neural network is slightly less transparent compared to a statistical analysis. As mentioned, cross-validation is the primary method to ensure that the training phase is conducted as intended.

Analysis of the final trajectory prediction performance is done by evaluating the performance metrics, summarized in Section 3.3.5. Since all the results are numerical, ROC curves will not be used. RMSE and MSE are therefore the leading measures to evaluate the performance. During the execution and hyperparameter tuning of the NN, the output variables from Table 4.3 will be directly evaluated. Only as a final validation, possibly comparing the results to those of related works, the actual metrics will be used. For each model iteration (including different features), the performance will be assesed by comparing the look-ahead time (Dependent variable)versus the performance metrics (Independent variables) which are measures as either RMSE or MSE. The baseline model serves as the 'ground truth' to which the remaining models are verified against.

Validation

Validation of the LSTM NN include performing the trajectory prediction with the selected air traffic features under different environmental conditions, such as region of application and timespan. For example, testing the model in a low and high traffic environment will reveal that the model works as intended. In contrast to a high traffic environment, the trajectory prediction in a low traffic environment is expected to not be significantly different to the baseline model. As the features and model have been selected for sector-independence, testing the model in a completely different sector should still reveal promising results. This is also true for seasonal effects, testing the model in different times of the year.

The post-analysis of the results can be related to the higher objective of this study. The benefits of improved trajectory prediction include increased airspace capacity. This can be quantified by calculating increased potential throughput by lowering safety margins taken to separate aircraft or by redistribution of aircraft in a sector, thus optimizing the demand and capacity balance.

Lastly, throughout the research a core design principle related to machine learning will be adhered to as much as reasonably possible. This involves the explainability of AI. This will be done by reporting on intermediate results and by visual analytics. This is vital if the knowledge gained from the research were to be used by controllers or pilots alike.

Limitations

At this stage of the research, the following limitations have been identified. As mentioned HMM, GMB, and LSTM are identified as most feasible algorithms. It is determined in Section 4.2 that LSTM is the most suitable of the three. However, this dependency on a single algorithm is a risk. The great advantage of machine learning is also the greatest risk in this case. This is the fact that the algorithms can identify patterns that are cognitively inconceivable and at a huge scale compared to possible analytical methods. This also means it is practically impossible to predict if the model will work and which exact adaptation to a model is the best performing. Based on previous experience in related work and general understanding of the model and the problem a choice is made. However, the risk remains that a HMM or GBM outperforms the LSTM.

Despite the feature selection and semi-verification of relationships between the features and the output variables, there remains the risk of not improving the accuracy of the trajectory prediction. In the case that the performance of the trajectory prediction is not improved, a detailed analysis will still be conducted. The expected knowledge gain from better understanding the relationship between the air traffic dynamics and the flown trajectory or anomalies to the filed flight plan will still be valuable. This is so because as ATM moves towards free routing and TBO, decentralized ATC will become more important. Any knowledge on how the local air traffic situation does or does not affect controller or pilot intent will contribute to this shift towards decentralized ATC. In fact, this could seen as an additional reason that on-board and decentralized ATM are not so dependent on surrounding traffic behavior.

5 Preliminary Analysis of Air Traffic Dynamic Features

This chapter goes into detail on the statistical preliminary analysis of the relationship between the air traffic dynamics features and the track deviation and delay. Frist, Section 5.1 will give insight into a key few aspects of the data representation prior to the computing of the features. Section 5.2 will give details on how each feature is calculated and defined. Then, in Section 5.3 the actual statistical analysis and evaluation is conducted, followed up by a more in depth discussion of some suprising results in Section 5.4

5.1. Data Representation

Prior to conducting the statistical analysis, some choices and pre-processing steps need to be taken. This includes devising the grid representation of the sector feature values, interpolating the raw flight points, and calculating the Kernel Density Estimation to reduce the discretization errors.

Grid Representation

Some of the air traffic dynamics features can be represented as scalar values unique to each aircraft. Is is thus possible to find relations between the individual dynamics and the flown trajectory. However, this would exclude the interaction effects of surrounding air traffic. Alternatively, it is possible to compute the air traffic dynamics feature value for an entire region, such as the EDUU UIR. The maximum distance between two point within EDUU is over 800 km. It is highly unlikely that air traffic dynamics between such a distance affects one another. A grid representation allows for the consideration of surrounding air traffic in computing air traffic dynamics features. This approach has previously been used during trajectory prediction based on weather information in various research [4, 27]. In order to capture the consideration horizon of pilot and controllers it makes sense to only consider a volume of space around the aircraft. The volume of airspace around an aircraft that will be monitored for air traffic and acted upon varies for a number of reasons. The vertical, lateral and longitudinal separation requirement could define the minimum grid size, however it is unlikely that aircraft will interact within the protected zone since it should never be more than 1 aircraft. The TCAS Traffic Advisory regions is expected to be a suitable minimum grid size. An visual overview of the TCAS protected volumes of airspace is shown in Figure 5.1 The TCAS display range is the maximum distance at which it is expected for a pilot to consider surrounding at traffic. However, for controllers this distance can be even greater and depends on the size of area of responsibility and personal controller behavior. Tuning the grid size so that it accurately represents this 'consideration horizon' will be important to capture air traffic dynamics effects.

Another consideration is error introduced due to discretization. To illustrate, if two aircraft are near each other but each in a different grid it would introduce an error. For that reason, *Kernel Density Estimation* (KDE) is applied to represent a distribution around the grid. See Figure 5.2 for a visual representation of the aircraft count in a grid for varying grid sizes. Notice the discretization effects at the boundaries of some grids.

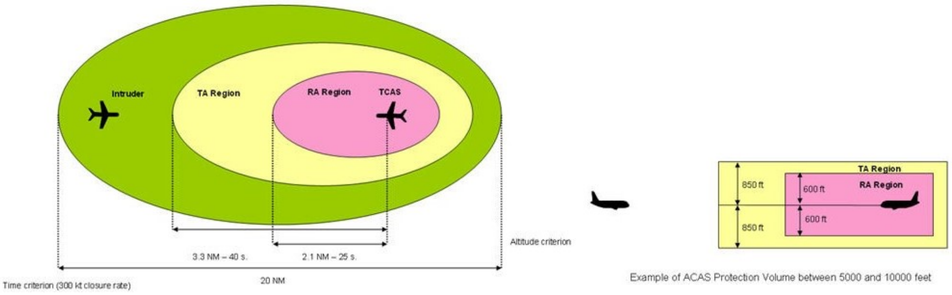
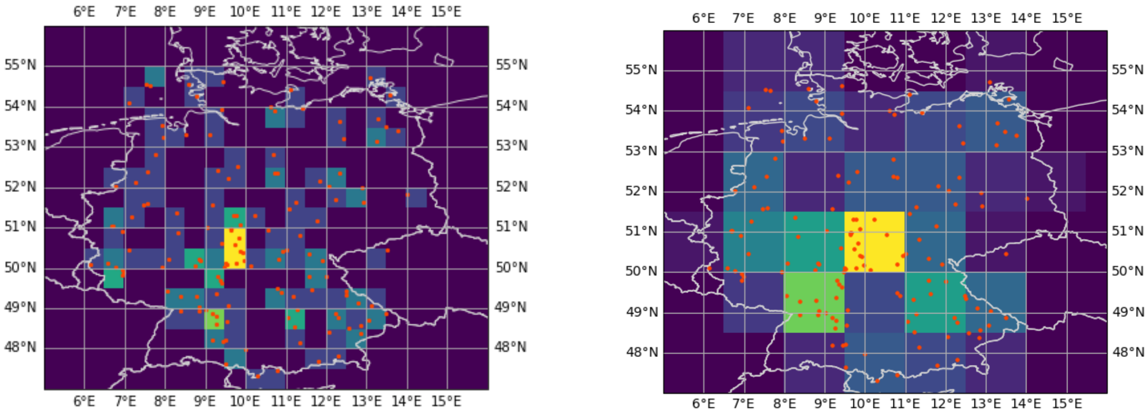


Figure 5.1: The TCAS protected volume of airspace for a typical aircraft. Note that the traffic advisory depends on time, so the distance is a function of the speed. Figure taken from Eurocontrol.



(a) 0.5 degree grid size

(b) 1.5 degrees grid size

Figure 5.2: ACC in a grid representation for two different grid sizes.

Kernel Density Estimation

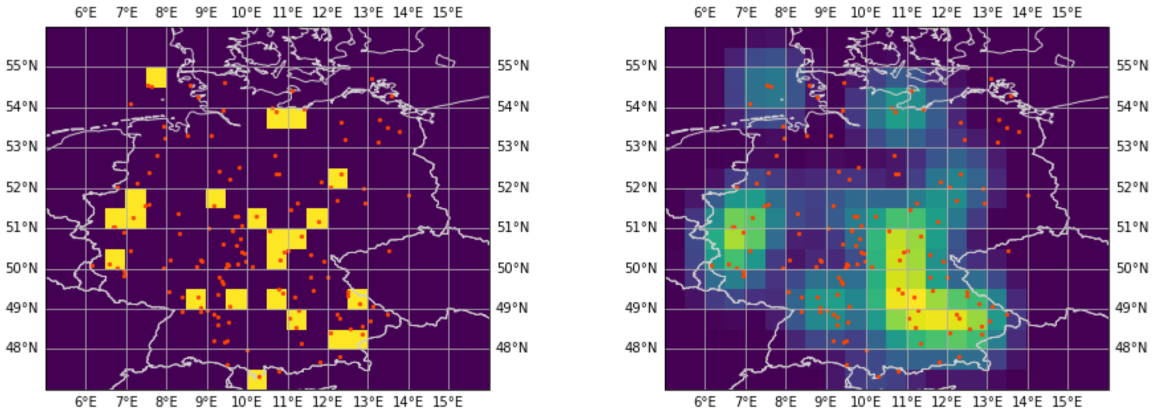
As mentioned, to represent some features more like a heatmap, KDE is applied. The bandwidth of the kernel should be carefully selected to represent realistic volumes of airspace that might have an effect on the trajectory. The KDE is implemented with a quartic kernel shape: Equation (5.1) represents the intensity at a point as a result of the initial intensity at the center point of the kernel that is within the selected kernel radius:

$$I = P\frac{15}{16}\left(1-\left(\frac{d}{h}\right)^2\right)^2 \tag{5.1}$$

Where, I is the (KDE) intensity, P is the local intensity (from raw data), d is point distance, h is the kernel radius. The effects from implementing the KDE on a 0.5 degree grid size with a 1.5 degrees kernel size is shown in Figure B.6.

Flight Point Interpolation

The raw data set contains roughly 30 data points per total trajectory. This means that only a few data points will span across the selected region. Also, all the time steps are different per flight. It is chosen to interpolate the flight points to allow a higher sampling rate of feature grid values along a trajectory. It is chosen to perform linear interpolation because most of the flights fly at a fairly constant heading angle, the turning radii are small compared to the scale of the sector, and if there is a slight curvature in the trajectory the flight point will most likely still fall in the right grid. Each variable is independently interpolated with respect to time. During the initial phase of the study, the size of the interpolation is fixed at 30 points with the selected region.



(a) Without KDE applied

(b) With KDE applied

Figure 5.3: AltCnt feature with and without KDE applied.

At a later stage, the air traffic dynamic features are calculated at constant interval time stamps. It is ensured that these time stamps are included in the interpolation to return accurate position data of flights. Figure 5.4 shows the results of a few interpolated flights. The colored dots represent the original flight points, the black dots represent the interpolated flight points. Notice that the interval of the black dots are not constant, to accommodate for the set time intervals, in this case every 5 minutes.

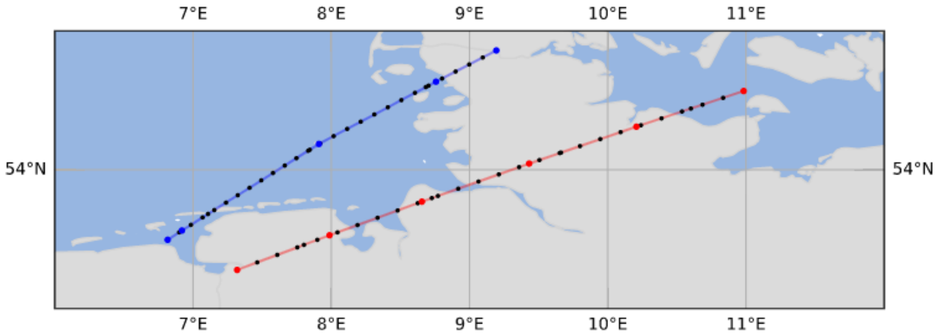


Figure 5.4: A flight showing the filed (blue) and actual (trajectory).

5.2. Data Processing

This section will cover the data processing steps taken to compute each air traffic feature from the air traffic models and also the metrics that will be used to compare the features against.

5.2.1. Independent Variables

The goal of the final predictive model is not to best predict an existing complexity model. The goal is to better predict aircraft trajectories. Section 3.3.5 covers the metrics that evaluate the output of the final trajectory prediction model. The statistical analysis must also be evaluated based on similar output metrics. It is expected that the relation between an exact spatial difference, either as polar coordinates or cross- and along-track deviation, and the selected features are highly dimensional and far too non-linear to capture with a statistical analysis. The closest variables that represents the predicted position or position error from the filed data points is the horizontal track deviation and the delay. Therefore we must consider differences between the filed and the actual trajectory in order to understand the air traffic dynamics and identify what features play a role in the flown trajectory. To evaluate the suitability of a feature in predicting the flown trajectory, the correlation is sought. In this statistical analysis, these differences are the metrics that are used to evaluate the features. These metrics are the delay accumulated while passing through the sector and the spatial deviation between the actual trajectory and the filed trajectory.

5.2.2. Delay

In this research delay is defined as the difference between the actual and filed duration that an aircraft takes to pass through a sector. This is done by comparing the filed and actual entry and exit times throughout the sector, see Equation (5.2). Many flights are already delayed before entering the region, taking the absolute difference between filed and actual exit times would thus not say anything about the trajectory anomalies that occurred while passing the region of interest. Moreover, some aircraft enter and exit the region multiple times when flying in the vicinity of the sector boundary. These cases are only considered if the filed flight plan and the actual flight points have the same amount of entry and exits. Otherwise it is not possible to distinguish which actual entry and exit logs correspond to those that are filed. This method of computing the delay means that negative delay is possible, meaning some aircraft need less time to pass through the region than filed. Although the general consensus may be that delay is worse than being ahead of schedule, this is not necessarily true since it is also disruptive to nominal operations. Also, there is no reason to assume that the 'negative delay' does not occur as a result of the air traffic dynamics. Therefore the 'negative delay' will be taken into account in further analysis. This definition of delay does leave some room for interpretation, which will discussed after the results are analysed.

$$\begin{aligned} \text{Duration} &= \text{Time of Entry} - \text{Time of Entry} \\ \text{Delay} &= \text{Duration}_{actual} - \text{Duration}_{filed} \end{aligned} \tag{5.2}$$

5.2.3. Track Deviation

Track deviation in this research is defined as the spatial difference between the filed trajectory and the actual flown trajectory. As mentioned in Section 3.3.5, there are multiple metrics for spatial errors. For the statistical analysis however it is expected that the trends between the cross- and along-track error and a single Euclidean distance in the horizontal plane will not be significantly different. However, the horizontal and vertical plane will be kept separated.

- **Instantaneous Track Deviation per Grid** At each time instant, the average and the sum of the track deviation for all flights per grid will be computed and compared to the air traffic dynamical feature at that grid at the same instance of time. This is done by computing the geodesic distance between each actual flight point and the nearest filed flight point and directly computing the difference in flight level. This method is time invariant. The reason for using the nearest filed flight point instead the same sequence number of flight point is because for many flights the starting and end point in a region to do not at all correspond. This can be seen in Figure 5.4. The distance is the geodesic distance, the shortest distance on the surface of an ellipsoidal model of the earth. The summed horizontal track deviation per grid at a certain time is seen in Figure B.7c.

- **Aggregated Track Deviation per Flight** The direct comparison of instantaneous track deviation will neglect any delayed effects. It is reasonable to assume that some effects of air traffic dynamics are measurable only a short while later in time. A pilot or controller could react to a situation, of which the effects are noticed with a certain delay. Therefore, the cumulative track deviation over a full trajectory is compared to the cumulative air traffic feature grid values encountered at each flight point. This approach is expected to take into account any delayed effects between air traffic dynamics and the track deviation. For each flight, the total aggregated distance is calculated by integrating the track deviation between the first and last point on the trajectory with respect to time. Since each trajectory is equally divided in a number of interpolated flight points, the integral is in fact equal to the sum of the individual track deviations over the whole trajectory.

5.2.4. Dependent Variables

The power of a NN is that it can identify and apply underlying relationships and patterns that we, humans, have not or cannot anticipate up front. So if an existing, highly processed, model were to serve as an input to a predictive model, it is possible that some of these relationships are missed. It therefore makes sense to consider the inputs to the existing model as individual values and not as a single aggregated value for complexity. The features are chosen in Section 4.1 and adapted to suit the context of this thesis.

Aircraft count

This straightforward feature counts the number of aircraft in each grid. See fig. B.1a for a visual representation at a time instant. For the delay and aggregated track deviation, the count in each grid is summed over the trajectories.

Density

Density is calculated by the Kernel Density Estimation. This is relevant because this is a spatial feature and the selected size of the area has a big effect on the value. See fig. B.1b for a visual representation at a time instant. For the delay and aggregated track deviation, the Density in each grid is summed over each trajectories.

Heading Change

This features results in two dependent variables per grid. First, the total count of aircraft making >15 degree heading change within 2 minute period. Second, the average heading change of aircraft making >15 degree heading change within 2 minute period. The two parameters: threshold for heading change; and time period should be tuned. The heading change count features is also calculated with KDE. See fig. B.2a, fig. B.2b, and fig. B.2c for a visual representation at a time instant. For the delay and aggregated track deviation, both feature values in each grid are summed over each trajectories.

The heading angles are calculated by solving the so called inverse geodesic problem. At each timestep, the coordinates of the consecutive points are used to compute the forward azimuth angle, which is the path direction from the first point. The azimuth angle corresponds to the bearing angle with respect to true north. The assumption is made that the aircraft do not experience drift from wind effects, and thus the bearing angle with respect to true north is assumed to equal the heading angle.

Note that the literature review suggested this feature be computed as a fraction, not a sum. It is chosen to take the sum in each grid because the original Dynamic Density feature was considered for the whole sector. A sum in that case correlates too much with the total aircraft count, thus a fraction is more representative. Since in a grid the amount of aircraft are much smaller, often 1 or 2, the law of small numbers would mean that a fraction of the total will easily be misinterpreted. For example, one out of 2 aircraft making a large heading change would result in a 0.5 value. However, 3/9 aircraft making a significant heading change is definitely more ’complex’, however the fraction in that case would only be 0.33. This is also true for the speed change and altitude change features.

Speed Change

This features results in two dependent variables per grid. First, the total count of aircraft with an airspeed change of >10 kts within a 2 minute period. Second, the average speed change of aircraft with an airspeed change of >10 kts within a 2 minute period. The two parameters: threshold for speed change; and time period should be tuned. The speed change count features is also calculated with KDE. See fig. B.3a, fig. B.3b, and fig. B.3c for a visual representation at a time instant. For the delay and aggregated track deviation, both feature values in each grid are summed over each trajectories. The speed at each point is computed by dividing the geodesic distance by the time difference between two flight points.

Flight Level Change

This features results in two dependent variables per grid. First, the summation of aircraft making >750 ft altitude change within 2 minute period. Second, the average altitude change of aircraft making >750 ft altitude change within 2 minute period. The two parameters: threshold for altitude change; and time period should be tuned. The speed change count features is also calculated with KDE. See fig. B.4a, fig. B.4b, and fig. B.4c for a visual representation at a time instant. For the delay and aggregated track deviation, both feature values in each grid are summed over each trajectories.

Minimum Distance

This feature results in three dependent variables per grid. The features equals the count of aircraft pairs at 3D Euclidean distance less than 5 NM, between 5-10 NM, and between 10-50 NM. Since the difference in altitude for en-route flights is relatively small, only the horizontal euclidean distance is considered. The distance between each aircraft pair equals the geodesic distance. This feature is split into three variable as this might shed light on what distance range might have an effect on an aircraft trajectory anomalies. Since this feature is a discrete count of occurrences per grid, it is reasonable to apply KDE. See fig. B.5a, fig. B.5b, fig. B.5c, fig. B.6a, fig. B.6b, fig. B.6c for a visual representation at a time instant.

Mean Separation

This features is simply the average distance between all aircraft pairs in the sector, calculated per grid. Since it already is an averaged variable, processing it by KDE is not suitable. See fig. B.7a for a visual representation at a time instant.

Standard Deviation of Aircraft Headings

This feature is computed by considering all aircraft in each grid and computing the standard deviation of the heading angles. Since this result is a aggregated value, the standard deviation, computing a KDE for continuity would not be suitable. See fig. B.7b for a visual representation at a time instant.

Size

This feature is a result from the aggregation of features over a whole trajectory. The aggregation is taken at constant flight instances, for example, every 5 minutes. The size counts the number of instances recorded for the aggregate features. For aircraft that spend little time in the selected region, the size is thus relatively small. The size thus relates to the duration flown inside the sector.

5.3. Statistical Data Analysis

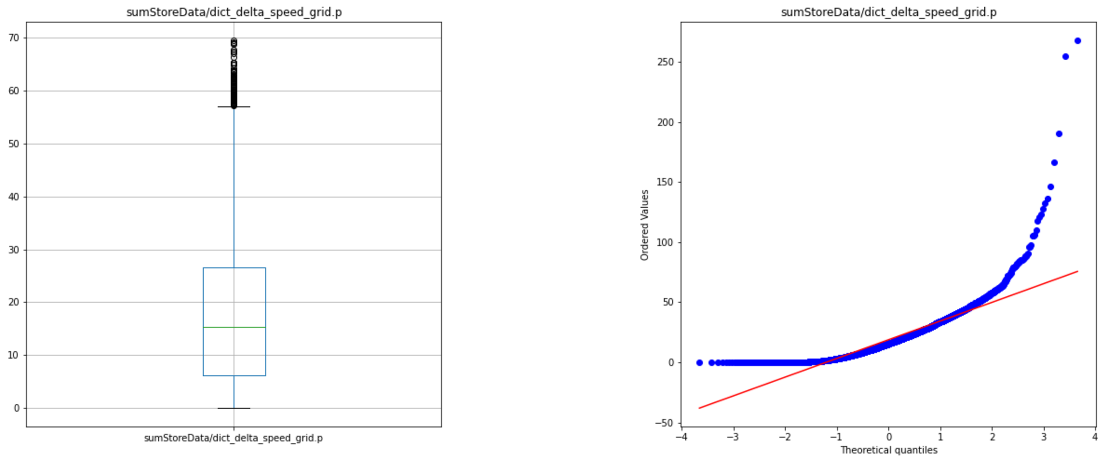
To analyze the relationship between the dependent variables and the independent variables, a statistical analysis is conducted. The choice of appropriate statistical test depends on the type of data. The first obvious distinction is that the input and output variables consist of numerical data: ratio or interval. This means that Pearson's correlation coefficient (parametric) and Spearman's rank coefficient (non-parametric) are the most suitable statistical tests. To asses which statistical test ought to be used, the following steps need to be taken.

Remove Noise

Each time instance produces a grid-representation of data. However, the majority of grid bins are not occupied by any aircraft nor hold any feature values. These grids carry no statistical information. The grids that contain no values of either independent or dependent variables are taken into consideration, also if the other respective variable has no information. In addition, outliers are removed from the data by disregarding data that is outside 3 standard deviations.

Test for Normality

Pearson's correlation coefficient is a parametric test, meaning that it works under the assumption that the variables are individually normally distributed (bivariate normality). If this assumption is not met, a non-parametric test should be conducted, the Spearman's rank coefficient test. Data can be tested for normality by graphical methods or formal statistical methods. Graphical methods include Q–Q plots, histograms or box plot. There are various formal statistical tests for normality. However, with very large sample sizes, formal statistical tests are not reliable.[52] Therefore, it is chosen to use the Q-Q plots in combination with box-plots to graphically confirm or deny the assumption of normality for each variable. In Figure 5.5, an example of the Q-Q plot and boxplot is shown of a feature that does not pass the normality test.



(a) Boxplot, showing skewness

(b) Q-Q plot, not passing the linearity condition

Figure 5.5: Tests of normality not passed for the average speed change above the thresholds.

In fact, with a selected grid size equal to half a degree on a geodetic coordinate system, none of the features pass the normality test with confidence. This is true for the aggregated and non-aggregated features. Three conclusions can be drawn from the data, as seen in Appendix B. Firstly, the spread of the data is so large that even after removing outliers, the spread is too large. This can be seen by the ourliers in the boxplots and in the divergent tails of many Q-Q plots. Secondly, some features are discrete and do not have enough observations to resemble any distribution. Third, several features have a one-sided distribution, meaning the mean is zero and the samples follow a normal-like distribution to one direction. This data cannot be classified as normal. Methods exists to solve this last issue but it is not deemed very beneficial for this application. Moreover, it is desirable to use the same statistical test for all features to best compare the features directly.

The third assumption for the Pearson test is that of linearity. The most practical method to asses this is visually with a scatter plot. However, since it is concluded that none of the features pass the normality test with confidence, the features do not need to be tested for linearity and the Pearson's correlation coefficient is not calculated.

Spearman's Coefficient of Rank Correlation

Spearman's coefficient of rank correlation is a non-parametric statistical test and provides a measure of how close two sets of rankings correlate with another. The Spearman coefficient (ρ) ranges between -1 and 1. Spearman's coefficient is suitable for both continuous and discrete ordinal data. This is relevant since the 'count' features are discrete ordinal variables. The threshold for rejecting the null-hypothesis lies at $p < \alpha = 0.02$. Usually, this threshold is 0.05, but the amount of data samples is so large that a higher threshold can be demanded. In contrast to the Pearson test, the Spearman coefficient is not a measure of linear association. The spearman coefficient assesses monotonic relationships. If one variable increases in value and so does the other (or the perfect inverse) there is a monotonic relationship. The Spearman coefficient is actually a

Pearsons coefficient between the rank variables and given in Equation (5.3) [1].

$$\rho = 1 - \frac{\mathrm{cov}(R_x, R_y)}{\sigma_x, \sigma_y} = 1 - \frac{6\sum d_i^2}{n(n^2-1)} \tag{5.3}$$

Where, $R_{x,y}$ denotes the rank variable, $\sigma_{x,y}$ is the standard deviation, d_i is the difference in paired ranks, and n is the number of pairs.

The results of the Spearman test and the corresponding p-value with a gridsize of 0.5 degrees on a geodetic coordinate system are given in Table 5.1 and Table 5.2. The value of h in the KDE is set at 1.5 degrees.

#	Feature	Track Deviation		FL Deviation	
		ρ	p-value	ρ	p-value
1	Average heading change	0.118	0.000	0.050	0.000
2	Average speed change	0.048	0.000	0.034	0.000
3	Average FL change	0.048	0.000	-0.019	0.009
4	Heading change count	0.060	0.000	0.030	0.000
5	Fl change count	0.063	0.000	-0.029	0.000
6	Speed change count	0.063	0.000	0.013	0.072
7	Standard deviation heading	0.286	0.000	-0.009	0.213
8	Mean separation	-0.080	0.000	0.001	0.864
9	Count 0-5NM	0.140	0.000	-0.008	0.286
10	Count 5-10NM	0.175	0.000	-0.016	0.024
11	Count 10-50NM	0.200	0.000	-0.011	0.145
12	Aircraft count	0.341	0.000	-0.014	0.062
13	Density KDE	0.540	0.000	0.005	0.130
14	Heading change count KDE	-0.178	0.000	0.008	0.129
15	Fl change count KDE	0.255	0.000	-0.007	0.106
16	Speed change count KDE	0.297	0.000	0.007	0.097
17	Count 0-5NM KDE	-0.215	0.000	-0.008	0.137
18	Count 5-10NM KDE	0.067	0.000	-0.002	0.658
19	Count 10-50NM KDE	0.475	0.000	0.004	0.238

Table 5.1: Spearman's coefficient of rank correlation for the individual features at 0.5 degree grid size

To study the sensitivity to the grid size, several tests are run at varying grid sizes. Due to the scale of the data, testing the data at a higher fidelity is not necessary. Any errors introduced by the interpolation is expected to be random and thus not affect the distribution with such large sample sizes. The remaining results are given in Appendix B. Note that for the aggregate features, an additional metric is present. This is the mean FL deviation aggregated over a trajectory, where the FL deviation per grid is the sum of all aircraft in that particular grid. The reason for adding this metric is to experiment if different aggregations lead to higher correlations, since the FL deviation correlations are consistently very weak and many statistically insignificant.

Cross-Correlation

Before the analysis can be conducted, one must test for cross-correlation to detect any features who's distribution are too similar. This is done for dimensionality reduction and to avoid bias in the model. The cross-correlation is tested by computing the normalized covariance matrix. So in fact, the cross-correlation is the Pearson's coefficient between variables. The Spearman rank is not suitable for this since then the rank determines the correlation, not the actual sample distribution. Figure 5.6 and Figure 5.7 shows the correlation matrix of the features and aggregated features respecitvely for a grid size of 0.5 degrees. The labels correspond to the numbering as in Table 5.1 and Table 5.2. The cross-correlation matrices for the remaining grid size are in Appendix B.

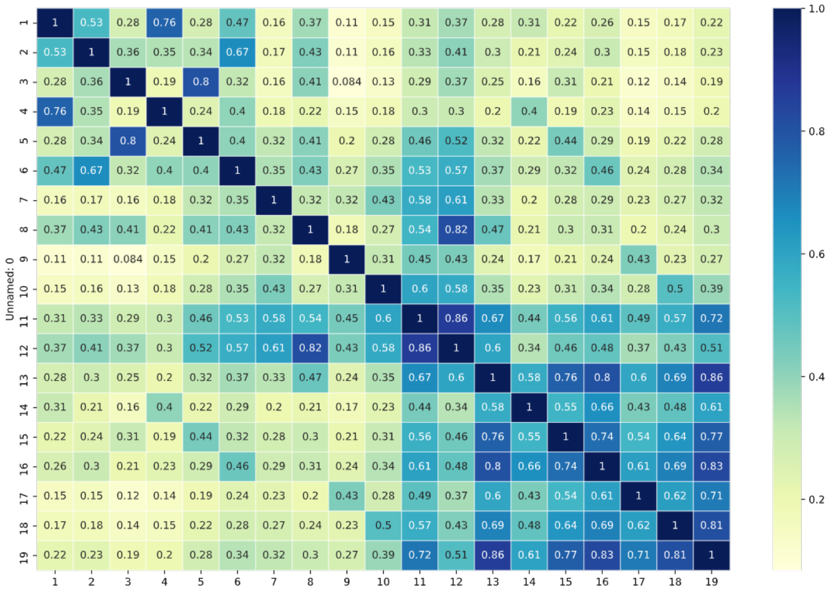
There is no fixed correlation coefficient threshold at which it is fair to say two variables are too similar in distribution. Some pairs will be evaluated:

- Heading change count and Average heading change: It is logical that these two correspond since these variables are adaptations of the same feature. The Spearman rank correlation is very different between

[1] https://statistics.laerd.com/statistical-guides/spearmans-rank-order-correlation-statistical-guide.php

#	feature	TD aggregated		FL deviation integral		FL deviation mean		Delay	
		ρ	p-value	ρ	p-value	ρ	p-value	ρ	p-value
19	Average heading change	0.376	0.000	0.057	0.000	0.078	0.000	-0.031	0.023
20	Size	0.547	0.000	0.020	0.142	0.032	0.018	-0.016	0.238
21	Average speed change	0.363	0.000	0.041	0.002	0.062	0.000	-0.031	0.024
22	Average FL change	0.142	0.000	-0.005	0.734	0.001	0.953	-0.114	0.000
23	Heading change count	0.222	0.000	0.043	0.002	0.053	0.000	-0.031	0.022
24	Fl change count	0.174	0.000	-0.032	0.017	-0.021	0.123	-0.114	0.000
25	Speed change count	0.303	0.000	0.014	0.294	0.035	0.011	-0.061	0.000
26	Standard deviation heading	0.320	0.000	0.001	0.942	0.019	0.153	-0.037	0.006
27	Mean separation	0.545	0.000	0.021	0.129	0.031	0.023	0.010	0.453
28	Count 0-5NM	0.182	0.000	-0.024	0.080	-0.012	0.395	-0.049	0.000
29	Count 5-10NM	0.253	0.000	-0.018	0.197	-0.003	0.843	-0.040	0.004
30	Count 10-50NM	0.361	0.000	-0.009	0.492	0.015	0.282	-0.056	0.000
31	Aircraft count	0.464	0.000	-0.004	0.760	0.013	0.334	-0.044	0.001
32	Density KDE	0.433	0.000	-0.003	0.831	0.020	0.146	-0.048	0.000

Table 5.2: Spearman's coefficient of rank correlation for the aggregated features at 0.5 degree grid size



Unnamed: 0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
1	1	0.53	0.28	0.76	0.28	0.47	0.16	0.37	0.11	0.15	0.31	0.37	0.28	0.31	0.22	0.26	0.15	0.17	0.22
2	0.53	1	0.36	0.35	0.34	0.67	0.17	0.43	0.11	0.16	0.33	0.41	0.3	0.21	0.24	0.3	0.15	0.18	0.23
3	0.28	0.36	1	0.19	0.8	0.32	0.16	0.41	0.084	0.13	0.29	0.37	0.25	0.16	0.31	0.21	0.12	0.14	0.19
4	0.76	0.35	0.19	1	0.24	0.4	0.18	0.22	0.15	0.18	0.3	0.3	0.2	0.4	0.19	0.23	0.14	0.15	0.2
5	0.28	0.34	0.8	0.24	1	0.4	0.32	0.41	0.2	0.28	0.46	0.52	0.32	0.22	0.44	0.29	0.19	0.22	0.28
6	0.47	0.67	0.32	0.4	0.4	1	0.35	0.43	0.27	0.35	0.53	0.57	0.37	0.29	0.32	0.46	0.24	0.28	0.34
7	0.16	0.17	0.16	0.18	0.32	0.35	1	0.32	0.32	0.43	0.58	0.61	0.33	0.2	0.28	0.29	0.23	0.27	0.32
8	0.37	0.43	0.41	0.22	0.41	0.43	0.32	1	0.18	0.27	0.54	0.82	0.47	0.21	0.3	0.31	0.2	0.24	0.3
9	0.11	0.11	0.084	0.15	0.2	0.27	0.32	0.18	1	0.31	0.45	0.43	0.24	0.17	0.21	0.24	0.43	0.23	0.27
10	0.15	0.16	0.13	0.18	0.28	0.35	0.43	0.27	0.31	1	0.6	0.58	0.35	0.23	0.31	0.34	0.28	0.5	0.39
11	0.31	0.33	0.29	0.3	0.46	0.53	0.58	0.54	0.45	0.6	1	0.86	0.67	0.44	0.56	0.61	0.49	0.57	0.72
12	0.37	0.41	0.37	0.3	0.52	0.57	0.61	0.82	0.43	0.58	0.86	1	0.6	0.34	0.46	0.48	0.37	0.43	0.51
13	0.28	0.3	0.25	0.2	0.32	0.37	0.33	0.47	0.24	0.35	0.67	0.6	1	0.58	0.76	0.8	0.6	0.69	0.86
14	0.31	0.21	0.16	0.4	0.22	0.29	0.2	0.21	0.17	0.23	0.44	0.34	0.58	1	0.55	0.66	0.43	0.48	0.61
15	0.22	0.24	0.31	0.19	0.44	0.32	0.28	0.3	0.21	0.31	0.56	0.46	0.76	0.55	1	0.74	0.54	0.64	0.77
16	0.26	0.3	0.21	0.23	0.29	0.46	0.29	0.31	0.24	0.34	0.61	0.48	0.8	0.66	0.74	1	0.61	0.69	0.83
17	0.15	0.15	0.12	0.14	0.19	0.24	0.23	0.2	0.43	0.28	0.49	0.37	0.6	0.43	0.54	0.61	1	0.62	0.71
18	0.17	0.18	0.14	0.15	0.22	0.28	0.27	0.24	0.23	0.5	0.57	0.43	0.69	0.48	0.64	0.69	0.62	1	0.81
19	0.22	0.23	0.19	0.2	0.28	0.34	0.32	0.3	0.27	0.39	0.72	0.51	0.86	0.61	0.77	0.83	0.71	0.81	1

Figure 5.6: The cross-correlation matrix for the features at a grid size of 0.5 degrees.

these variables. So, especially with a varying grid size this feature is deemed different enough to maintain.

- Average FL change and Fl change count: This is the same as above. For the individual features, FL change count has higher rank correlation, it is decided to remove the average FL change.

- Aircraft count is highly cross correlated with both density KDE and mean separation. Since Density shows a much higher Spearman rank coefficient and mean separation is not cross correlated with density KDE, it is decided to remove aircraft count. The aircraft count makes sense to resemble the density since the area in which aircraft are counted is the same area as used to compute the density.

- Both the KDE and non-KDE Count of 10-50NM separation show relatively high correlation with several other features. This is especially true for the KDE variable. The reason to not remove this feature is that is shows high rank correlation with the track deviation. However, the highest cross-correlation is with

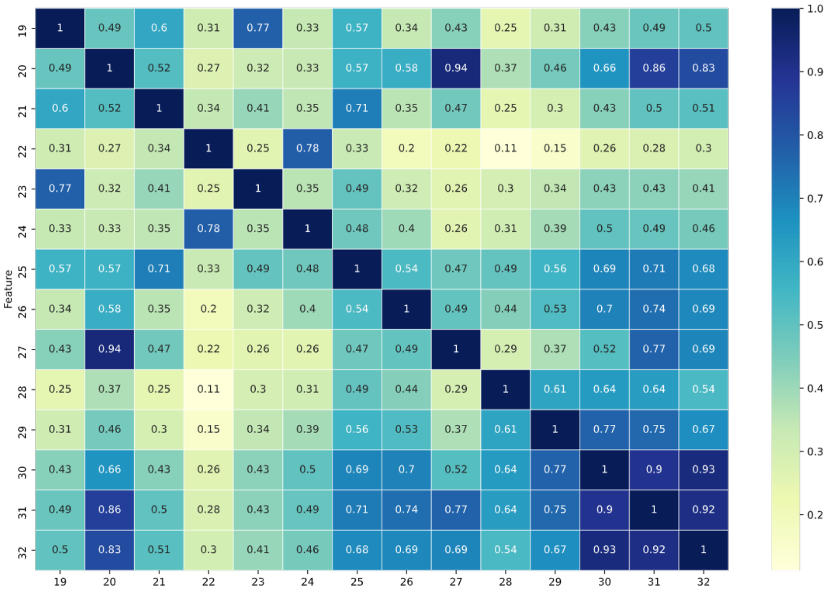


Figure 5.7: The cross-correlation matrix for the aggregated features at a grid size of 0.5 degrees.

density and speed change count which both also have a reasonably high rank coefficient. Therefore it is decided to remove the Count of 10-50NM KDE feature.

- The features for which a KDE has been estimated show high levels of cross correlation (bottom right). This can be explained. The track deviation value is zero for grids that don't contain flights. However, the KDE features can include feature values in grids that don't contain any flights because they are near to the edge of surrounding grids. The grids that are zero for non-KDE features, but contain a value for KDE features are correlated with the KDE distribution that is applied. So grids that were zero are allocated a value, hence the rank is increased. However, these effects should be neglected since it is not a causal relationship but a statistical flaw that introduces bias. Therefore the KDE correlations will be excluded from the evaluation of the feature rank correlation.

For the aggregated features:

- Size and Mean Separation: These two features are highly correlated. Initially, this may seem somewhat surprising because the size refers to the number of flight points that are taken into account for the aggregation. This would suggest there is a relationship between the time spend inside the sector and the mean separation. This can be explained because flights that spend a lot of time inside the sector will often fly through the center of the sector, and tend to have a lower mean separation. The size feature will be removed because it is (intuitively) less related to the air traffic dynamics.
- Lastly, the Count at 10-50NM, Aircraft count, and Density KDE are all highly correlated. This is a logical conclusion since all these features depend on the separation between aircraft in a enclosed space (the grid). If the grid size is within the 10-50NM range, it is essentially the same measure. It is expected that this result will vary for different grid size but not significantly. The cross-correlation is the highest between Density KDE and the other two features. Therefore the total Aircraft count and Count at 10-50NM are removed.

5.3.1. Evaluation

To draw conclusions from the statistical analysis, a few aspects must be kept in consideration. Firstly, correlation does not imply causality. Secondly, the law (of better said, fallacy) of small numbers. Moreover, the bivariate testing is not indicative of possible trivariate relationships between the metrics and two (or more) features. Relating to the last remark, the purpose of this statistical analysis is to learn about the possible effect of air traffic dynamics on trajectory anomalies and provide a knowledge basis on which to perform data-driven trajectory prediction using LSTM NNs.

To start with the FL Deviation. None of the computed features have a convincing statistical monotonic relation with FL deviation. 14 of the 19 variables for individual grid features do not pass the null-hypothesis. The aggregated features lead to similar outcomes, both the integrated and mean FL deviations. This suggests two things: the FL deviations are so small that there are no significant correlations and the FL deviations are too random to prove any significance. It suggests that ATC and pilots barely resort to vertical manoeuvres in order to de-conflict complex air traffic situations. This is as expected as it is a know controller strategy. This has interesting implications for the trajectory prediction. Since there are not even one or two features that can be singled out, FL prediction is a high uncertainty for the trajectory prediction. It is theoretically possible that highly non-linear multivariate relations exists between the air traffic dynamical features and FL deviations, but deemed unlikely. It is suggested that in the very least the vertical element of the trajectory prediction is kept separated from the horizontal component. If the computational efforts are too high, the first adaptation will be to remove the vertical component from the trajectory prediction.

Next, the delay metric. Quite surprisingly, there is practically no correlation between the air traffic dynamical features and the delay. Several adaptations of this metric have also been tested, including: The absolute value of the delay to try and find any correlation with only the magnitude of delay but also with that metric no noteworthy correlations are found; The positive delay values (so only aircraft taking longer to pass through the region), but again no noteworthy correlations are found. This can mean a few things. Either the metric calculation is faulty and the metric does not represent the conventional concept of delay. Delay is the difference in duration to pass through the sector between the filed and actual flight points. The edges of the sector are not straight lines, as as seen in Figure 5.4, the entry and exit points do not correspond. So naturally, the exit and entry times are very different. This is a major source of noise. Alternatively, the conclusion is that for en-route traffic, delay is not significantly (linearly or low order non-linearly) affected by the surrounding air traffic dynamics. This last conclusion does not correspond with the expectation. However, it is important to note that the metrics of the statistical analysis, track deviation, FL deviation, and delay, are not inputs nor direct outputs of the LSTM NN. This is further discussed in the discussion below.

As for the horizontal track deviation, there are some interesting and promising results. The individual features per grid show that the standard deviation of heading angle and density have the highest rank correlation. Generalizing, the KDE features have higher correlation, but are ignored as mentioned before. Especially for the aggregated features, significant correlations can be observed with the track deviation. The difference between individual grid feature values and aggregated values suggests that there is a strong delay effect between grid value and track deviation. This is precisely the reason why the aggregated features have been included in this phase of the study. So, the knowledge gained from this phase of the study means that it is important to include time varying effects in the trajectory prediction. That means that at a certain time, the trajectory prediction will depend on grid values at past time instances. This does put more confidence in the choice of machine learning algorithm, since the LSTM NN is adapted for time-series and takes into account past information.

Lastly, comparing the results for different grid sizes suggests that the grid size of 1.5 degrees has consistently higher rank correlations with track deviation, for both the individual as the aggregated features. 1.5 degrees is in the order of 160-170 km. This is larger than expected and and is because of the increased number of flights in a grid, increasing the value of many complexity feature values. This can be confirmed by the increasing cross-correlation at large grid sizes. This is further discussed in the discussion below.

On a final note, the output variables of the LSTM NN and this statistical analysis are not identical. This means that one must be careful drawing definitive conclusions from this phase of the study about the trajectory prediction. Nevertheless, a overview of most promising features is given in Table 5.3

5.4. Discussion and Lessons Learned

To try and find correlations with the delay, is possible to filter the data further and isolate only the flights that enter without a delay and observe the results. Also, the opposite can be done, to investigate if the behavior of flights is different when the flight have a delay prior to entering the region. Possibly a flight 'speeds up' and 'cuts corners' when it has a existing delay which it tries to minimize. However, these approaches tend toward

Feature	**Remarks**
Density	KDE
Standard deviation of heading angle	-
Mean Separation	Whole sector correlation is higher than that of 10-50NM separation. So tuning of gridsize is required.
Average speed change	Tuning of threshold parameters required, difference with speed change count not apparent, can be interchanged
Average heading change	Tuning of threshold parameter required
FL change count	KDE; Tuning of threshold parameter required
Count 10-50NM separation	Tune parameter based on grid size (2-3x grid size)

Table 5.3: Air Traffic Dynamics features in order of highest likelihood to have significant effect on the flown trajectory.

confirmational bias and have not been pursued. The Eurocontrol complexity metrics for ANSP benchmarking identified the German UIR as one of the most complex of all European regions. This verifies that the cause of this lack of correlation is not because the region is not complex enough. [2] These observations on delay are not necessarily problematic for the performance of the trajectory prediction since the primary prediction metric will be the spatial coordinates at time-steps. (4D trajectory). The delay is derived from the trajectory prediction. The cross-correlation between track deviation and delay is 0.16 and statistically significant, which does suggest that there is a relationship.

On another note, at this stage the feature values are calculated by summing the individual aircraft feature values in each grid. The alternative is taking the mean of the individual feature values in each grid. However, during initial testing with small grid sizes the mean value was quite meaningless because of the law of small numbers: the number of aircraft in a grid was too small for the mean to statistically represent anything of value. However, it can be seen that the rank correlations are higher for large grid sizes. Since the feature values are summed in a grid, there is increasing cross-correlation with the number of samples in a grid. This problem will be attempted to be solved by normalizing the feature values with respect to the grid size. Either way, the sensitivity to the grid size which as been observed must be analysed by means of a sensitivity study as part of the trajectory prediction.

The air traffic dynamical features that have been chosen in Section 4.1 which have not yet been tested include: Lyapunov Exponents, Convergence-Divergence Rates, Conflict predicted, and the Eurocontrol Complexity metrics. Since the preliminary statistical study does suggest that there are relationships between the air traffic dynamics features and the flown trajectory, it is more effective to first move forward with the trajectory prediction. Once the concept of the trajectory prediction has been proven effective, the remaining features can be calculated. Since the Lyapunov Exponents are a unique and intrinsic measure of complexity, the priority will be in modelling the Lyapunov Exponents, then the Convergence-Divergence Rates, then the Conflict Predicted, and lastly the Eurocontrol Complexity Metrics.

The statistical analysis was performed on data from one day. For robustness and verification of these conclusions, multiple sample days in different months should be analysed. This should also include days that are not congested at all. This is very important because on those days it it more likely that the pilot has higher freedom to choose its route instead of having to follow strict procedures and commands by ATC during 'busy' days. It might therefore be a better capture of the true air traffic dynamics.

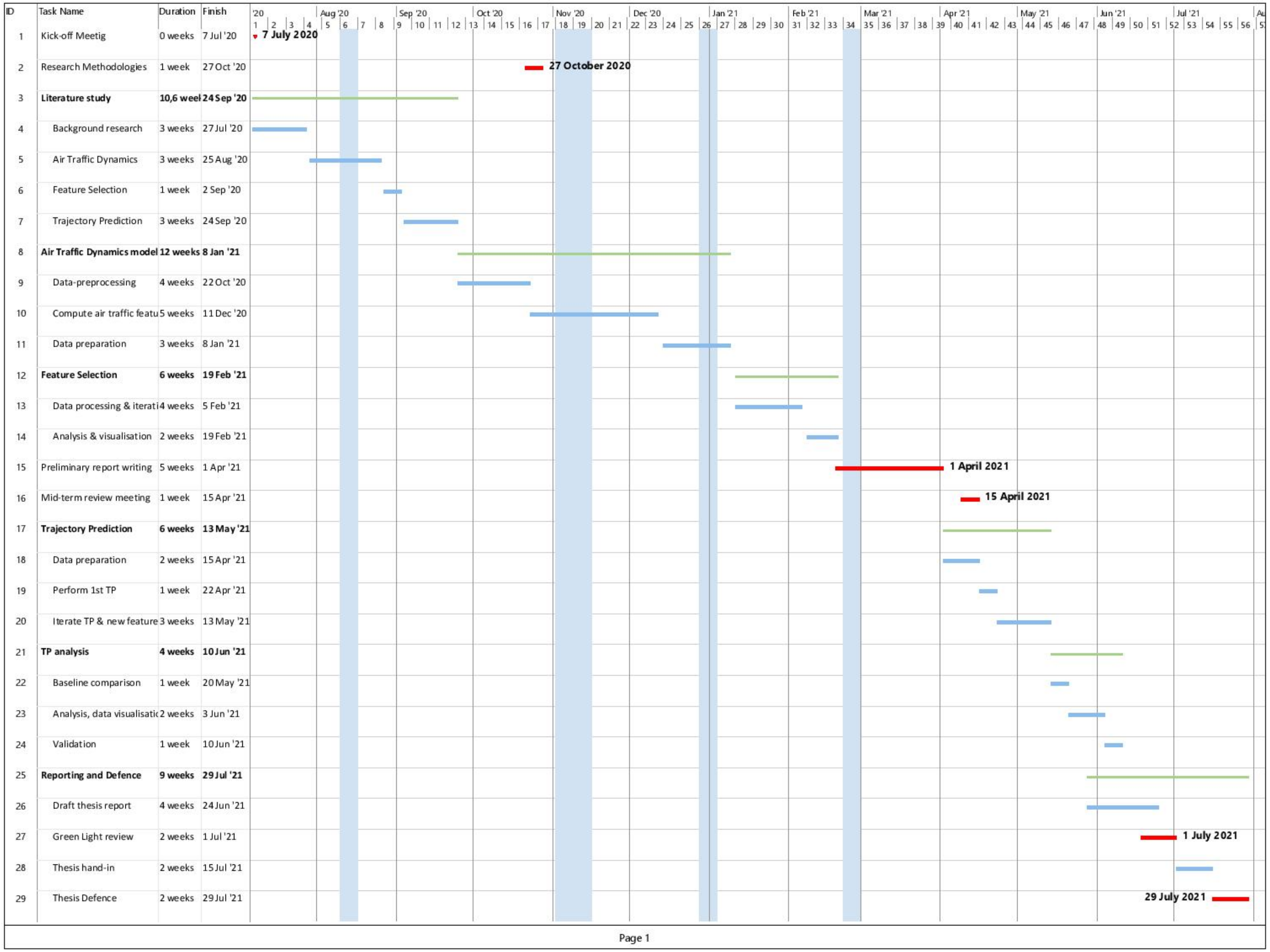
Since the uneven edges of the sector introduce a lot of noise and inconsistency with the data, it is chosen to extend the sector edges. This will also allow for a longer look-ahead time.

6
Scheduling and Planning

For the planning of the remained of the thesis project, we must consider all the major phases that are to be conducted. An overview of the phases is found in the Work Flow Diagram in Figure 2.2 and includes the execution steps as follows:

1. **Data Pre-processing** This phase is completed.
2. **Air Traffic Dynamic Feature Selection** This phase is nearing completion and the working principle has been proven. However, the following features remain to be computed and implemented in the statistical analysis:
    - Conflict predicted
    - Convergence rate
    - Eurocontrol Complexity Metrics
    - Lyapunov Exponent
3. **Data Preparation for TP** The steps needed to be taken in this phase are covered in Section 4.3.
4. **Trajectory Prediction and Baseline Comparison** This phase is the main experimental phase of the study. Once the preparation is done, this phase only consists of training the NN with the various inputs and testing the results.
5. **VV and Result Analysis** This very important phase will conclude the content of this Thesis project and draw conclusions on the results.

The conduct of the remained of this thesis will not be strictly in the same order as the phases mentioned above. With the current features, the data will be prepared for a trial trajectory prediction experiment. This will verify that the current plan and steps included are adequate and lead to some results that can be evaluated. Once the concept has been proven successful, the remaining features will be computed in order as given above. This will be conducted depending on the initial results and scope. If the set of features at each state provides sufficient results it will be chosen to emphasize on the analysis of those features. The planning is made with respect to time in the Gantt chart in Figure 6.1.



ID	Task Name	Duration	Finish
1	Kick-off Meetig	0 weeks	7 Jul '20
2	Research Methodologies	1 week	27 Oct '20
3	**Literature study**	**10,6 weel**	**24 Sep '20**
4	Background research	3 weeks	27 Jul '20
5	Air Traffic Dynamics	3 weeks	25 Aug '20
6	Feature Selection	1 week	2 Sep '20
7	Trajectory Prediction	3 weeks	24 Sep '20
8	**Air Traffic Dynamics model**	**12 weeks**	**8 Jan '21**
9	Data-preprocessing	4 weeks	22 Oct '20
10	Compute air traffic featu	5 weeks	11 Dec '20
11	Data preparation	3 weeks	8 Jan '21
12	**Feature Selection**	**6 weeks**	**19 Feb '21**
13	Data processing & iterati	4 weeks	5 Feb '21
14	Analysis & visualisation	2 weeks	19 Feb '21
15	Preliminary report writing	5 weeks	1 Apr '21
16	Mid-term review meeting	1 week	15 Apr '21
17	**Trajectory Prediction**	**6 weeks**	**13 May '21**
18	Data preparation	2 weeks	15 Apr '21
19	Perform 1st TP	1 week	22 Apr '21
20	Iterate TP & new feature	3 weeks	13 May '21
21	**TP analysis**	**4 weeks**	**10 Jun '21**
22	Baseline comparison	1 week	20 May '21
23	Analysis, data visualisatic	2 weeks	3 Jun '21
24	Validation	1 week	10 Jun '21
25	**Reporting and Defence**	**9 weeks**	**29 Jul '21**
26	Draft thesis report	4 weeks	24 Jun '21
27	Green Light review	2 weeks	1 Jul '21
28	Thesis hand-in	2 weeks	15 Jul '21
29	Thesis Defence	2 weeks	29 Jul '21

Figure 6.1: Gantt chart of entire Thesis planning

7 Conclusion

The objective of this thesis is to to improve the accuracy of medium- to long-term flight trajectory predictions by incorporating a data-driven model that encompasses the dynamics of the air traffic situation. In previous research, the features associated with air traffic dynamics have not been included in trajectory predictions. It will be a novelty to do so. However, the available literature on air traffic dynamics is primarily focused on the demand and capacity balance and air traffic complexity for controller workload modelling.

This thesis will be conducted in two phases that divide this research. The first phase includes generating and verifying a model for air traffic dynamics that relate to the flown trajectory. In this research the air traffic dynamical models based on Dynamic Density for air traffic complexity will be taken as a starting point. A statistical analysis is conducted to select features of this model that will be passed on to the second phase of this study. The second phase of this study includes predicting the 4D trajectory by including the features from the air traffic dynamics model. This is done by applying the Long Short-Term Memory Neural Network, which is specifically adapted for time-series prediction. A baseline comparison will be conducted that does not include any air traffic dynamics features in order to validate the effect of the features.

The preliminary results from the first phase of this thesis suggest that the horizontal track deviation correlate with some features from the Dynamic Density air traffic complexity model. The delay in duration of passing though the selected region (Karlsruhe UAC) as well as the Flight Level deviation do not have a statistically significant relationship with the selected dynamics features. However, since the output variables of the statistical analysis are not identical to the desired output of the trajectory prediction there remains an uncertainty. Moreover, there are some remaining and promising features that still need to be modelled. It can be concluded that there is a high probability that the selected features will have a beneficial effect to the performance of the trajectory prediction and therefore it is reasonable to begin the execution of the second phase of this thesis project. It is decided to first move forward with the trajectory prediction and following the proof of concept include the remaining air traffic dynamics features.

This preliminary report summarizes the first phase of this thesis project and includes a proposal of the method to adapt existing air traffic dynamics models and use these in a machine learning trajectory prediction. A planning is made that outlines the steps to be taken and estimates the time that is required to successfully conduct the research.

A

Air Traffic Comlexity factors

A (incomplete) list of unique air traffic complexity factors. Note that for each factor, many variations are possible and have been used in previous research. Where, for example, the sector geometry can be expressed by the number of sides or by a equation expressed in terms of major axis length and aspect ratio. Alternatively, the number of path changes can be a count of path changes when the heading changes by 5°, 10°, or 15°and the amount of altitude change for it to count as a path change can also vary. This list thus gives an overview of broad, undefined factors that need specification but can each contribute to the air traffic complexity.

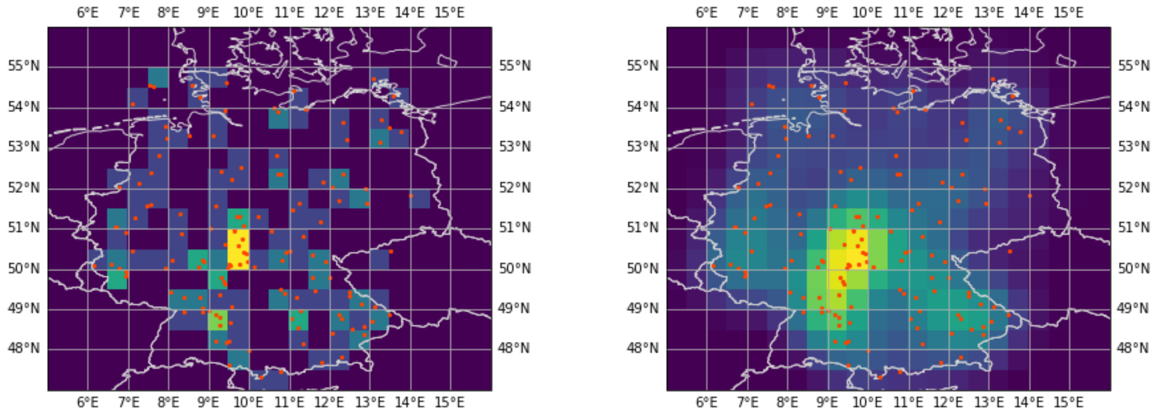
1. Number of aircraft
2. Aircraft density or traffic volume
3. Aircraft handled in prior time interval (e.g., last hour)
4. Number of arrivals
5. Number of departures
6. Number of emergencies
7. Number of special flights
8. Coordination
9. Traffic mixture (arrivals, departures, and over flights)
10. Number of airport terminals
11. Traffic distribution
12. Staffing
13. Weather conditions
14. Equipment status
15. Number of communications with aircraft
16. Number of communications with other sectors
17. Presence of conflicts
18. Number of path changes
19. Preventing conflicts (crossing or overtake)
20. Number of handoffs and printouts
21. Handling pilot requests
22. Traffic flow structure
23. Clustering of aircraft
24. Control adjustments involved in merging and spacing
25. Mixture of aircraft types
26. Combing and descending aircraft flight plans
27. Number of intersecting flight paths
28. Number of required procedures
29. Number of military flights
30. Airline hub location
31. Weather and its severity
32. Aircraft routing
33. Special use airspace
34. Sector geometry
35. Sector size
36. Requirements for longitudinal and lateral spacing
37. Radar coverage
38. Frequency congestion
39. Number of altitudes used

B

Intermediate Results Statistical Analysis

This appendix includes the results of the statistical analysis, including box plots, Q-Q plots, Spearman rank correlation coefficients for all the features at varying grid sizes, and lastly, the correlation matrices for all grid sizes.

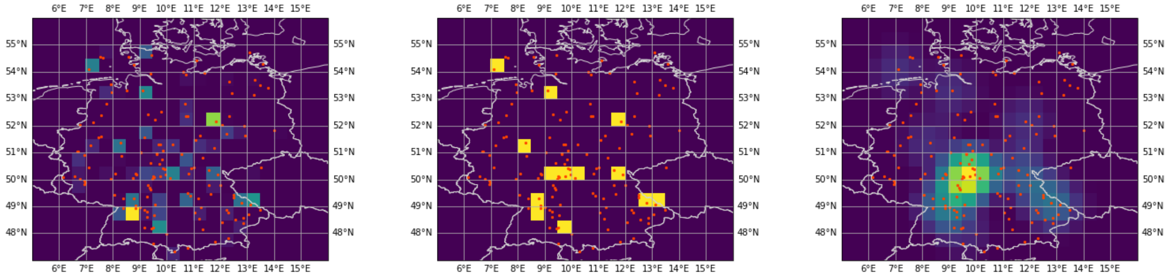
Visualisations of the air traffic dynamics features



(a) Aircraft count per grid at 12:00:00 02-09-2018.

(b) Density KDE per grid at 12:00:00 02-09-2018.

Figure B.1: Features at each time instant for each grid.

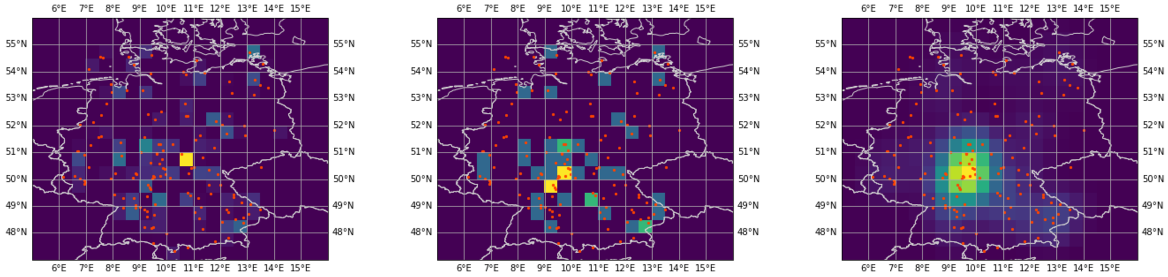

6°E 7°E 8°E 9°E 10°E 11°E 12°E 13°E 14°E 15°E
55°N 54°N 53°N 52°N 51°N 50°N 49°N 48°N

(a) Average heading angle change per grid at 12:00:00 02-09-2018.

(b) Heading angle count per grid at 12:00:00 02-09-2018.

(c) Heading angle count with KDE per grid at 12:00:00 02-09-2018.

Figure B.2: Heading change features at each time instant for each grid.

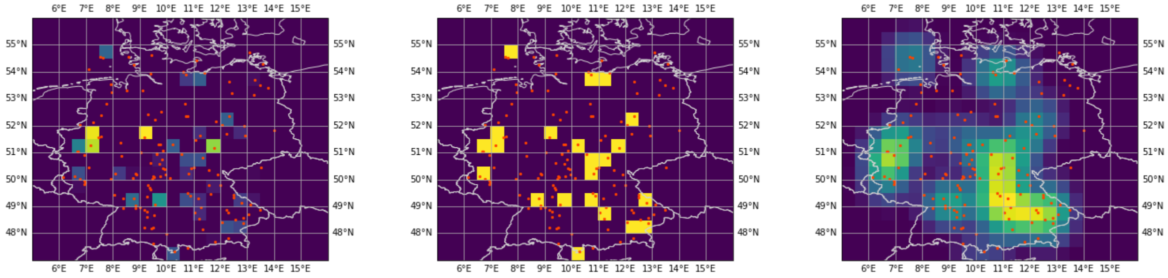

6°E 7°E 8°E 9°E 10°E 11°E 12°E 13°E 14°E 15°E
55°N 54°N 53°N 52°N 51°N 50°N 49°N 48°N

(a) Average speed change per grid at 12:00:00 02-09-2018.

(b) Speed change count per grid at 12:00:00 02-09-2018.

(c) Speed change count with KDE per grid at 12:00:00 02-09-2018.

Figure B.3: Speed change features at each time instant for each grid.

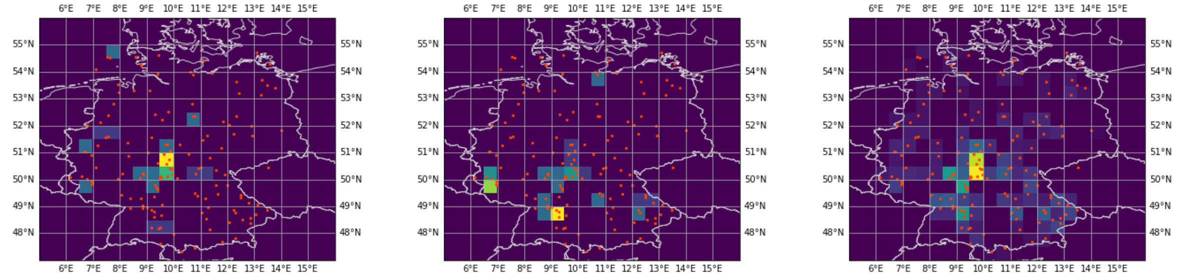

6°E 7°E 8°E 9°E 10°E 11°E 12°E 13°E 14°E 15°E
55°N 54°N 53°N 52°N 51°N 50°N 49°N 48°N

(a) Average FL change per grid at 12:00:00 02-09-2018.

(b) FL angle count per grid at 12:00:00 02-09-2018.

(c) FL angle count with KDE per grid at 12:00:00 02-09-2018.

Figure B.4: FL change features at each time instant for each grid.

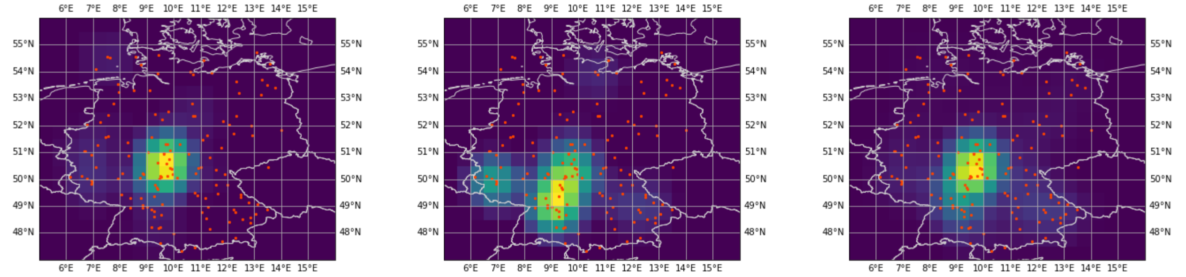


(a) Count of 0-5NM separation pairs per grid at 12:00:00 02-09-2018.

(b) Count of 5-10NM separation pairs per grid at 12:00:00 02-09-2018.

(c) Count of 10-50NM separation pairs per grid at 12:00:00 02-09-2018.

Figure B.5: Count of minimum separation pairs at each time instant for each grid.

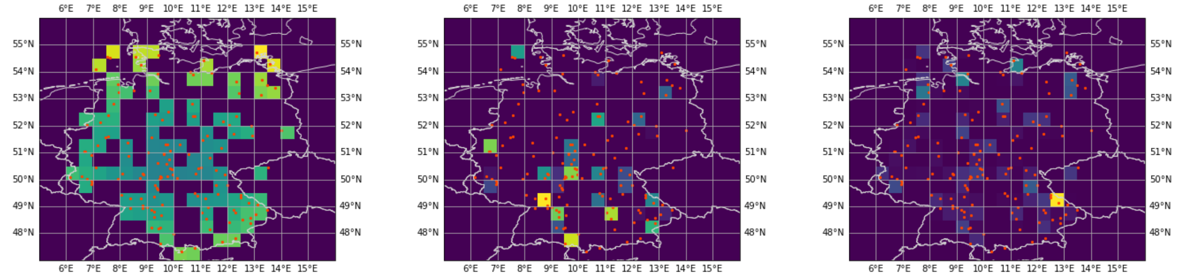


(a) KDE of count of 0-5NM separation pairs per grid at 12:00:00 02-09-2018.

(b) KDE of count of 5-10NM separation pairs per grid at 12:00:00 02-09-2018.

(c) KDE of count of 10-50NM separation pairs per grid at 12:00:00 02-09-2018.

Figure B.6: KDE of count of minimum separation pairs at each time instant for each grid.



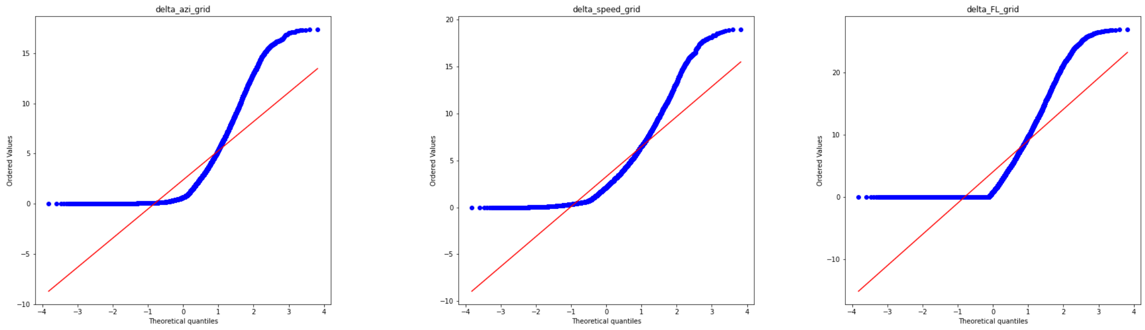
(a) Sum of mean separation for all aircraft per grid at 12:00:00 02-09-2018.

(b) Standard deviation of all heading angle of flights per grid at 12:00:00 02-09-2018.

(c) Sum of track deviation of flights per grid at 12:00:00 02-09-2018.

Figure B.7: Features at each time instant for each grid.

Tests of Normality

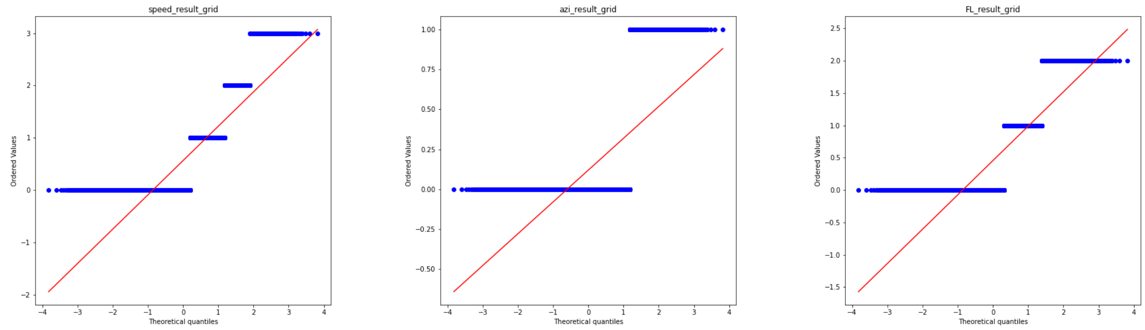


(a) Average heading change Q-Q plot.

(b) Average speed change Q-Q plot.

(c) Average FL change Q-Q plot.

Figure B.8: Tests of Normality for individual features.

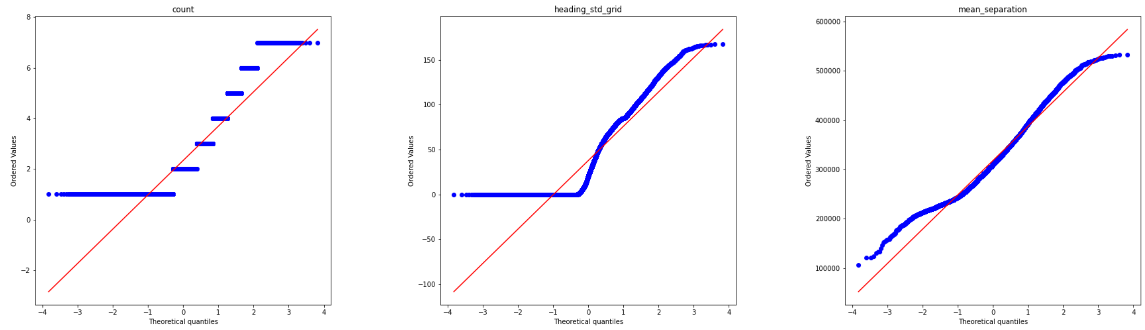


(a) Speed change count Q-Q plot.

(b) Heading change count Q-Q plot.

(c) FL change count Q-Q plot.

Figure B.9: Tests of Normality for individual features.

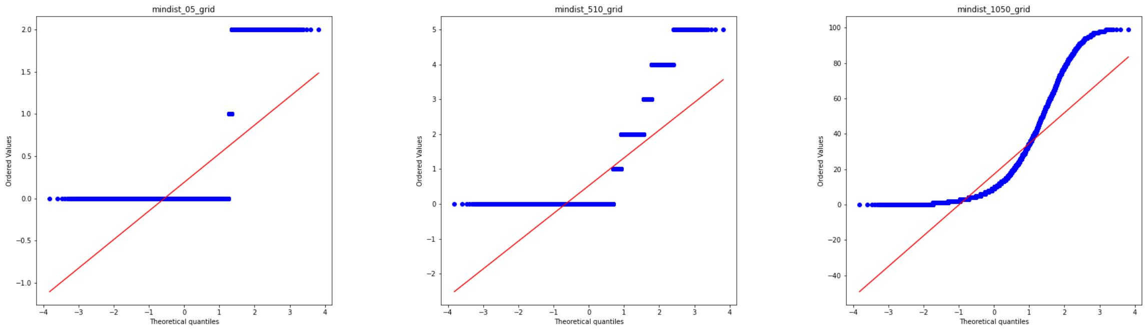


(a) Aircraft count Q-Q plot.

(b) Standard deviation of heading Q-Q plot.

(c) Mean separation Q-Q plot.

Figure B.10: Tests of Normality for individual features.

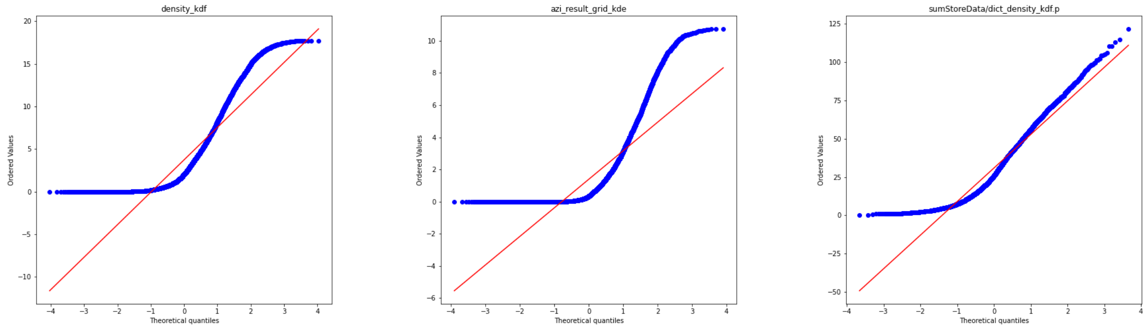


(a) Count of 0-5NM separation Q-Q plot.

(b) Count of 5-10NM separation Q-Q plot.

(c) Count of 10-50NM separation Q-Q plot.

Figure B.11: Tests of Normality for individual features.

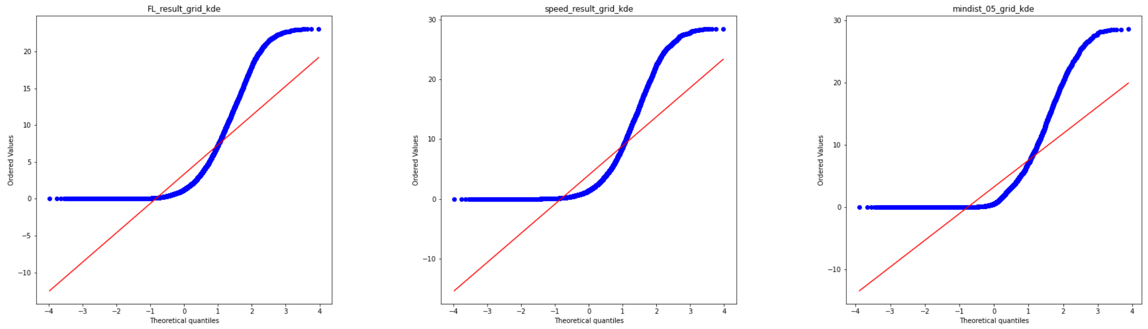


(a) KDE of Density Q-Q plot.

(b) KDE of heading change count Q-Q plot.

(c) Aggregated KDE of density Q-Q plot.

Figure B.12: Tests of Normality for individual and aggregated features.

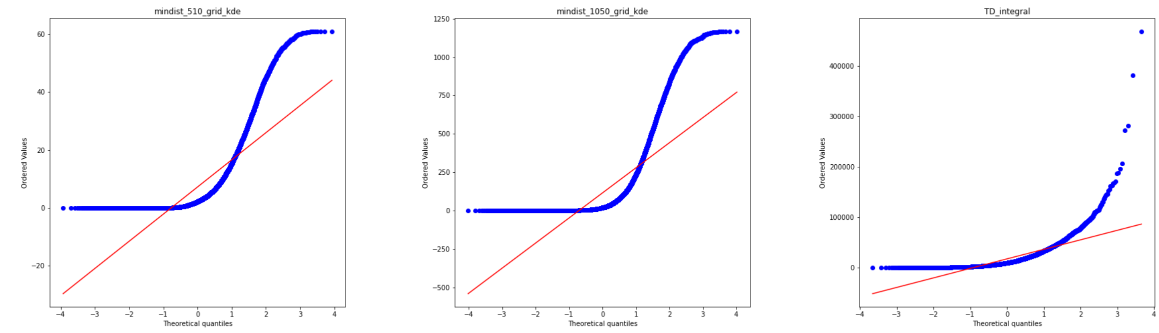


(a) KDE of FL change ount Q-Q plot.

(b) KDE of speed change count Q-Q plot.

(c) KDE of 0-5NM separation count Q-Q plot.

Figure B.13: Tests of Normality for individual features.

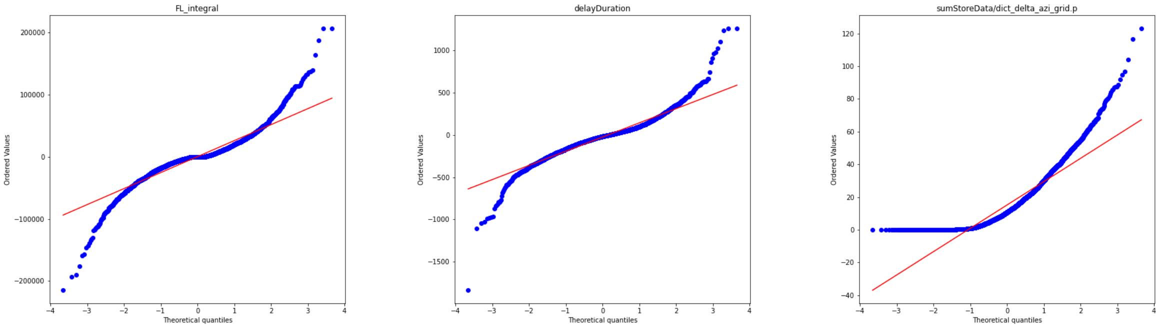


(a) KDE of 5-10NM separation count Q-Q plot.

(b) KDE of 10-50NM separation count Q-Q plot.

(c) Track Deviation Q-Q plot.

Figure B.14: Tests of Normality for individual features.

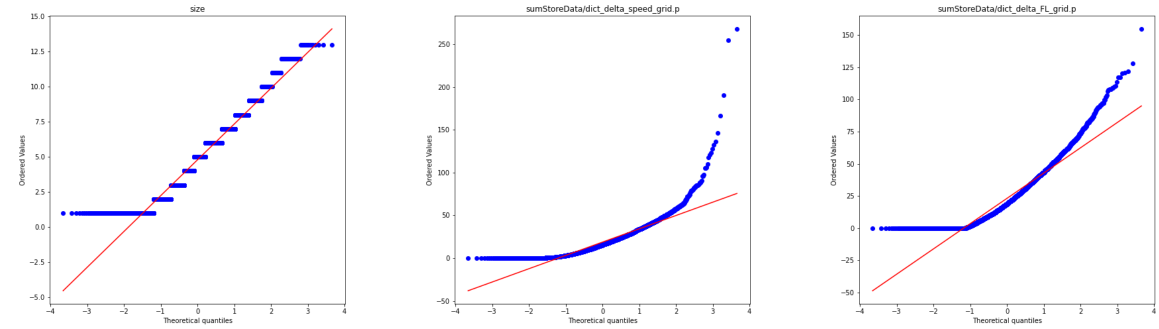


(a) FL deviation Q-Q plot.

(b) Delay Q-Q plot.

(c) Average heading change Q-Q plot.

Figure B.15: Tests of Normality for aggregated features.

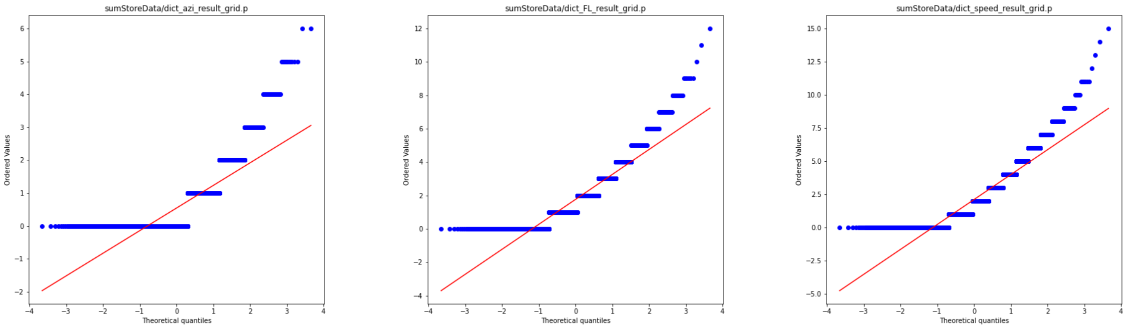


(a) Trajectory size Q-Q plot.

(b) Average speed change Q-Q plot.

(c) Average Fl change Q-Q plot.

Figure B.16: Tests of Normality for aggregated features.

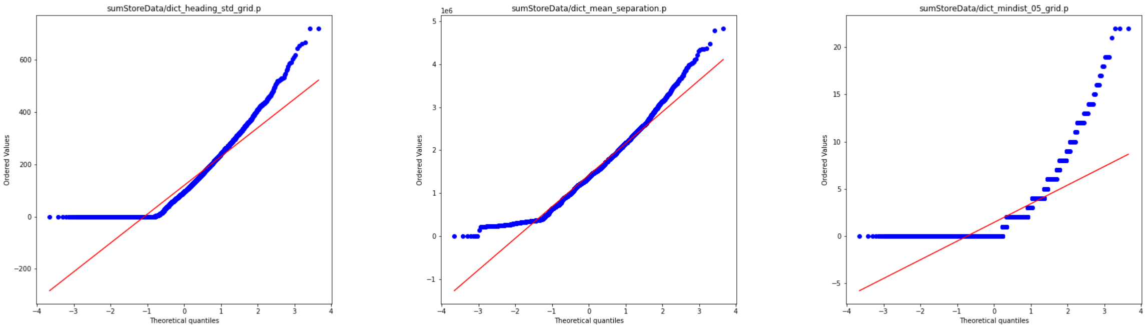


(a) Heading change count Q-Q plot.

(b) Fl change count Q-Q plot.

(c) Speed change count Q-Q plot.

Figure B.17: Tests of Normality for aggregated features.

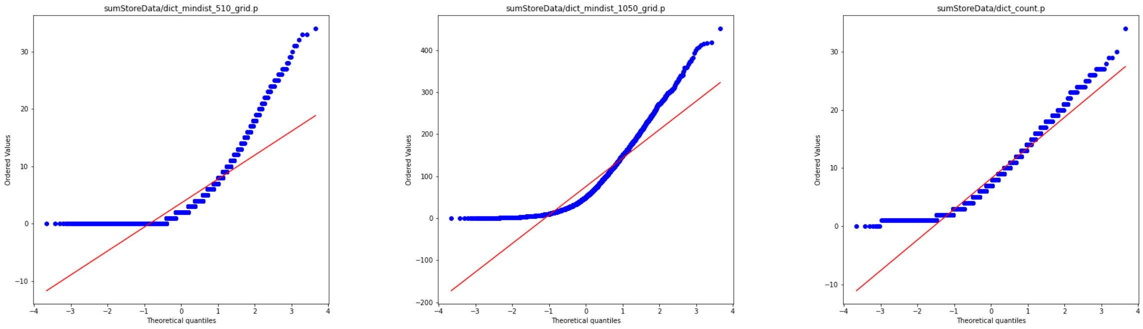


(a) Standard deviation of heading angle Q-Q plot.

(b) Mean separation Q-Q plot.

(c) Count of 0-5NM separation Q-Q plot.

Figure B.18: Tests of Normality for aggregated features.



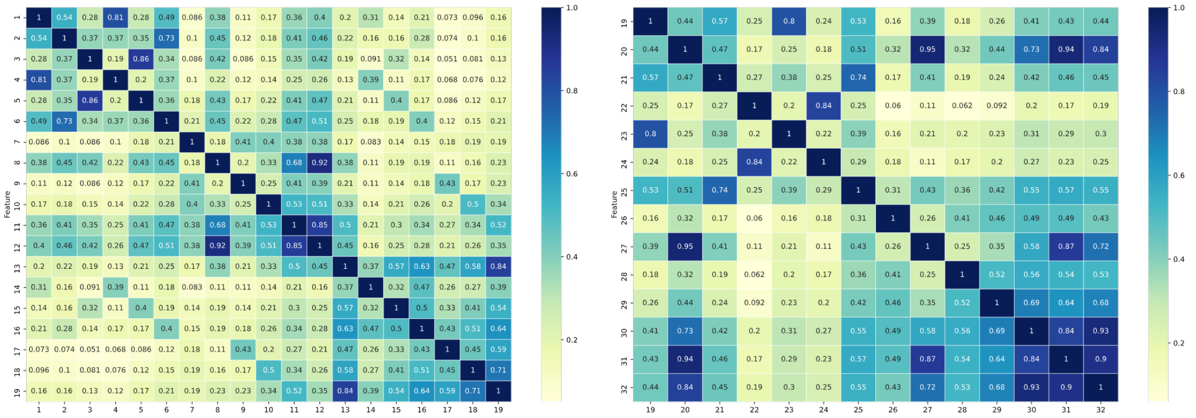
(a) Count of 5-10NM separation Q-Q plot.

(b) Count of 10-50NM separation Q-Q plot.

(c) Aircraft count Q-Q plot.

Figure B.19: Tests of Normality for aggregated features.

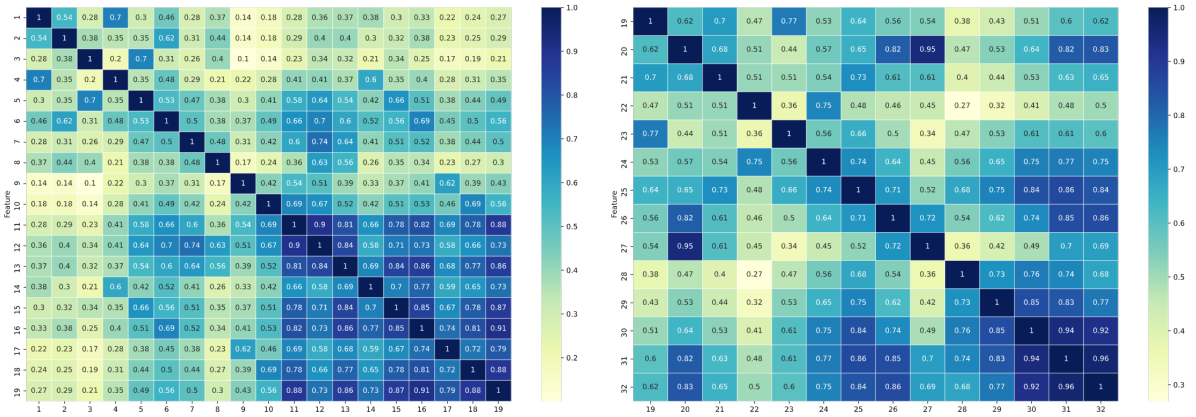
Cross Correlation Matrices



(a) The individual features

(b) The aggregated features

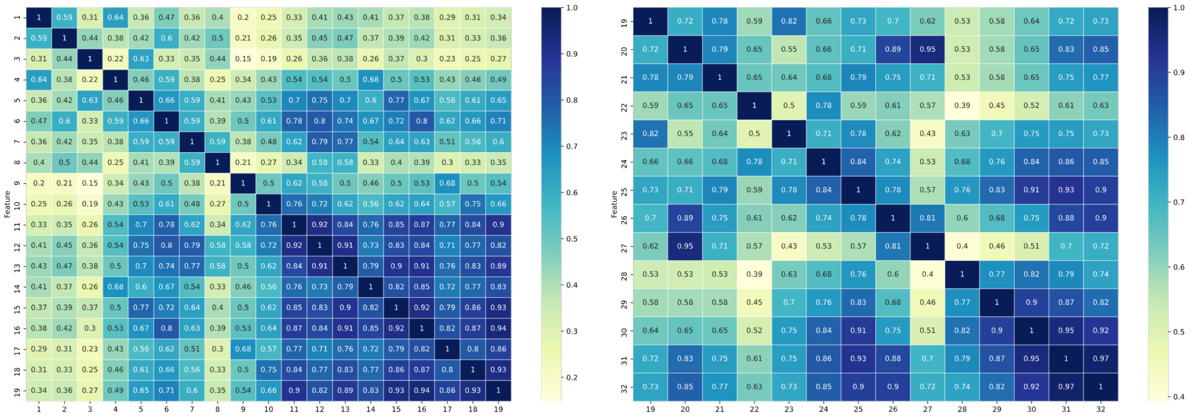
Figure B.20: Cross-correlation matrices of the features with 0.25 degree grid size.



(a) The individual features

(b) The aggregated features

Figure B.21: Cross-correlation matrices of the features with 1 degree grid size.



(a) The individual features

(b) The aggregated features

Figure B.22: Cross-correlation matrices of the features with 1.5 degree grid size.

Spearman's Coefficient of Rank Correlation Tables

#	Feature	Track Deviation		Flight Level Deviation	
		ρ	p-value	ρ	p-value
1	Average heading change	0.076	0.000	0.052	0.000
2	Average speed change	0.011	0.105	0.037	0.000
3	Average FL change	-0.007	0.297	-0.024	0.000
4	Heading change count	0.024	0.000	0.033	0.000
5	Fl change count	-0.007	0.254	-0.030	0.000
6	Speed change count	-0.001	0.831	0.018	0.005
7	Standard deviation heading	0.107	0.000	-0.008	0.213
8	Mean separation	-0.033	0.000	0.005	0.451
9	Count 0-5NM	0.086	0.000	-0.004	0.529
10	Count 5-10NM	0.050	0.000	-0.008	0.216
11	Count 10-50NM	0.045	0.000	-0.011	0.078
12	Aircraft count	0.182	0.000	-0.010	0.115
13	Density KDE	0.386	0.000	0.003	0.237
14	Heading change count KDE	-0.519	0.000	0.006	0.167
15	Fl change count KDE	-0.083	0.000	-0.015	0.000
16	Speed change count KDE	-0.029	0.000	0.001	0.682
17	Count 0-5NM KDE	-0.528	0.000	-0.012	0.009
18	Count 5-10NM KDE	-0.242	0.000	-0.006	0.093
19	Count 10-50NM KDE	0.304	0.000	0.002	0.321

Table B.1: Spearman's coefficient of rank correlation for the individual features at 0.25 degree grid size

#	Feature	Track Deviation		Flight Level Deviation	
		ρ	p-value	ρ	p-value
1	Average heading change	0.232	0.000	0.051	0.000
2	Average speed change	0.158	0.000	0.041	0.000
3	Average FL change	0.163	0.000	-0.030	0.002
4	Heading change count	0.154	0.000	0.034	0.000
5	Fl change count	0.241	0.000	-0.040	0.000
6	Speed change count	0.239	0.000	0.013	0.166
7	Standard deviation heading	0.441	0.000	-0.014	0.141
8	Mean separation	-0.237	0.000	-0.009	0.336
9	Count 0-5NM	0.221	0.000	-0.007	0.464
10	Count 5-10NM	0.305	0.000	-0.018	0.067
11	Count 10-50NM	0.469	0.000	-0.008	0.401
12	Aircraft count	0.547	0.000	-0.016	0.101
13	Density KDE	0.763	0.000	0.002	0.748
14	Heading change count KDE	0.149	0.000	0.021	0.010
15	Fl change count KDE	0.528	0.000	-0.009	0.189
16	Speed change count KDE	0.557	0.000	0.009	0.221
17	Count 0-5NM KDE	0.119	0.000	0.002	0.830
18	Count 5-10NM KDE	0.344	0.000	-0.003	0.695
19	Count 10-50NM KDE	0.671	0.000	0.001	0.910

Table B.2: Spearman's coefficient of rank correlation for the individual features at 1 degree grid size

#	Feature	Track Deviation		Flight Level Deviation	
		ρ	p-value	ρ	p-value
1	Average heading change	0.319	0.000	0.060	0.000
2	Average speed change	0.230	0.000	0.039	0.002
3	Average FL change	0.234	0.000	-0.021	0.093
4	Heading change count	0.267	0.000	0.044	0.000
5	Fl change count	0.407	0.000	-0.036	0.004
6	Speed change count	0.408	0.000	0.007	0.584
7	Standard deviation heading	0.501	0.000	-0.028	0.024
8	Mean separation	-0.372	0.000	-0.027	0.030
9	Count 0-5NM	0.306	0.000	0.001	0.921
10	Count 5-10NM	0.427	0.000	-0.018	0.154
11	Count 10-50NM	0.633	0.000	-0.009	0.473
12	Aircraft count	0.694	0.000	-0.016	0.206
13	Density KDE	0.807	0.000	0.008	0.319
14	Heading change count KDE	0.460	0.000	0.020	0.041
15	Fl change count KDE	0.699	0.000	0.007	0.408
16	Speed change count KDE	0.717	0.000	0.011	0.213
17	Count 0-5NM KDE	0.408	0.000	0.007	0.510
18	Count 5-10NM KDE	0.575	0.000	0.002	0.789
19	Count 10-50NM KDE	0.729	0.000	0.005	0.514

Table B.3: Spearman's coefficient of rank correlation for the individual features at 1.5 degree grid size

#	feature	TD aggregated		FL deviation integral		FL deviation mean		Delay	
		ρ	p-value	ρ	p-value	ρ	p-value	ρ	p-value
19	Average heading change	0.383	0.000	0.063	0.000	0.083	0.000	-0.025	0.066
20	size	0.547	0.000	0.020	0.142	0.032	0.018	-0.016	0.238
21	Average speed change	0.354	0.000	0.042	0.002	0.064	0.000	-0.026	0.052
22	Average FL change	0.078	0.000	-0.006	0.657	0.000	0.981	-0.112	0.000
23	Heading change count	0.207	0.000	0.055	0.000	0.062	0.000	-0.028	0.037
24	Fl change count	0.089	0.000	-0.023	0.093	-0.015	0.268	-0.105	0.000
25	Speed change count	0.289	0.000	0.027	0.050	0.047	0.001	-0.052	0.000
26	Standard deviation heading	0.241	0.000	-0.007	0.589	0.011	0.433	-0.032	0.020
27	Mean separation	0.551	0.000	0.022	0.113	0.031	0.021	0.013	0.338
28	Count 0-5NM	0.154	0.000	-0.011	0.439	-0.001	0.917	-0.046	0.001
29	Count 5-10NM	0.236	0.000	-0.010	0.454	0.003	0.836	-0.032	0.020
30	Count 10-50NM	0.387	0.000	-0.008	0.575	0.016	0.234	-0.052	0.000
31	Aircraft count	0.507	0.000	0.007	0.601	0.022	0.102	-0.026	0.051
32	Density KDE	0.453	0.000	-0.006	0.650	0.014	0.306	-0.044	0.001

Table B.4: Spearman's coefficient of rank correlation for the aggregated features at 0.25 degree grid size

#	feature	TD aggregated		FL deviation integral		FL deviation mean		Delay	
		ρ	p-value	ρ	p-value	ρ	p-value	ρ	p-value
19	Average heading change	0.399	0.000	0.063	0.000	0.082	0.000	-0.014	0.313
20	size	0.547	0.000	0.020	0.142	0.032	0.018	-0.016	0.238
21	Average speed change	0.411	0.000	0.041	0.003	0.059	0.000	-0.034	0.012
22	Average FL change	0.293	0.000	-0.009	0.523	-0.006	0.641	-0.102	0.000
23	Heading change count	0.256	0.000	0.034	0.013	0.047	0.001	-0.017	0.203
24	Fl change count	0.295	0.000	-0.022	0.102	-0.009	0.501	-0.094	0.000
25	Speed change count	0.341	0.000	0.011	0.426	0.033	0.016	-0.059	0.000
26	Standard deviation heading	0.422	0.000	-0.001	0.933	0.016	0.227	-0.038	0.005
27	Mean separation	0.550	0.000	0.020	0.134	0.031	0.023	0.013	0.344
28	Count 0-5NM	0.227	0.000	-0.014	0.286	-0.002	0.875	-0.052	0.000
29	Count 5-10NM	0.280	0.000	-0.011	0.427	0.008	0.569	-0.038	0.005
30	Count 10-50NM	0.343	0.000	-0.006	0.679	0.018	0.191	-0.057	0.000
31	Aircraft count	0.436	0.000	-0.002	0.888	0.017	0.204	-0.047	0.001
32	Density KDE	0.431	0.000	-0.001	0.936	0.021	0.116	-0.048	0.000

Table B.5: Spearman’s coefficient of rank correlation for the aggregated features at 1.0 degree grid size

#	feature	TD aggregated		FL deviation integral		FL deviation mean		Delay	
		ρ	p-value	ρ	p-value	ρ	p-value	ρ	p-value
19	Average heading change	0.422	0.000	0.047	0.001	0.066	0.000	-0.026	0.056
20	size	0.547	0.000	0.020	0.142	0.032	0.018	-0.016	0.238
21	Average speed change	0.443	0.000	0.029	0.034	0.045	0.001	-0.038	0.005
22	Average FL change	0.368	0.000	-0.006	0.659	-0.002	0.857	-0.091	0.000
23	Heading change count	0.299	0.000	0.022	0.111	0.040	0.003	-0.036	0.007
24	Fl change count	0.353	0.000	-0.015	0.275	0.004	0.770	-0.079	0.000
25	Speed change count	0.368	0.000	0.003	0.802	0.025	0.065	-0.052	0.000
26	Standard deviation heading	0.466	0.000	-0.007	0.605	0.011	0.430	-0.038	0.005
27	Mean separation	0.553	0.000	0.018	0.194	0.028	0.039	0.010	0.478
28	Count 0-5NM	0.255	0.000	-0.007	0.624	0.010	0.460	-0.042	0.002
29	Count 5-10NM	0.303	0.000	-0.011	0.436	0.008	0.534	-0.044	0.001
30	Count 10-50NM	0.347	0.000	-0.005	0.718	0.020	0.145	-0.049	0.000
31	Aircraft count	0.437	0.000	-0.004	0.747	0.017	0.204	-0.042	0.002
32	Density KDE	0.436	0.000	0.000	0.993	0.023	0.087	-0.047	0.001

Table B.6: Spearman’s coefficient of rank correlation for the aggregated features at 1.5 degree grid size

Bibliography

[1] A. Nuic. User Manual for the Base of Airfract Data (BADA). Technical report, Eurocontrol, 2009. URL `www.eurocontrol.inthttps://www.eurocontrol.int/eec/gallery/content/public/document/eec/report/2009/003_BADA_3_7_User_manual.pdf`.

[2] ACE Working Group on Complexity. Complexity Metrics for ANSP Benchmarking Analysis. Technical Report April, Eurocontrol, 2006.

[3] Petar Andraši, Tomislav Radišić, Doris Novak, and Biljana Juričić. Subjective air traffic complexity estimation using artificial neural networks. *Science in Traffic & Transportation*, 31(4):377–386, 2019. ISSN 03535320. doi: 10.7307/ptt.v31i4.3018.

[4] Samet Ayhan and Hanan Samet. Aircraft trajectory prediction made easy with predictive analytics. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 13-17-Augu:21–30, 2016. doi: 10.1145/2939672.2939694.

[5] B. Hilburn. Cognitive Complexity in Air Traffic Control - A Literature Review. Technical report, Eurocontrol Experimental Centre, 2004.

[6] Sun Choi, Young Jin Kim, Simon Briceno, and Dimitri Mavris. Prediction of weather-induced airline delays based on machine learning algorithms. In *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, volume 2016-Decem. Institute of Electrical and Electronics Engineers Inc., 12 2016. ISBN 9781509056002. doi: 10.1109/DASC.2016.7777956.

[7] DART. Data--Driven Aircraft Trajectory Prediction Exploratory Research. 2017.

[8] A. M.P. de Leege, M. M. van Paassen, and M Mulder. A machine learning approach to trajectory prediction. In *AIAA Guidance, Navigation, and Control (GNC) Conference*, 2013. ISBN 9781624102240. doi: 10.2514/6.2013-4782. URL `http://arc.aiaa.org`.

[9] C A Dek. *Predicting 4D Trajectories of Aircraft using Neural Networks and Gradient Boosting Machines.* PhD thesis, Delft University of Technology.

[10] Daniel Delahaye and Stephane Puechmorel. Air traffic complexity: towards intrinsic metrics. *3rd USA/Europe Air Traffic Management R&D Seminar*, (June):1–11, 2000. URL `http://tinyurl.com/affmbc6`.

[11] Daniel Delahaye and Stéphane Puechmorel. 4D Trajectories Complexity Metric Based on Lyapunov Exponents. 2011. URL `https://hal-enac.archives-ouvertes.fr/hal-01319023`.

[12] D. Scarlatti; P. Costas; E.Casado. Evaluation and Validation of Algorithms for Single Trajectory Prediction. pages 1–56, 2018.

[13] EUROCONTROL. Monthly Network Operations Report - Analysis September 2018. Technical Report September, EUROCONTROL, 2018.

[14] Federal Aviation Administration. The Future of the NAS The Future of the NAS Contents Letter from Assistant Administrator for NextGen Introduction. Technical report, 2016. URL `https://www.faa.gov/nextgen/media/futureofthenas.pdf`.

[15] Esther Calvo Fernández, José Manuel Cordero, George Vouros, Nikos Pelekis, Theocharis Kravaris, Harris Georgiou, Georg Fuchs, Natalya Andrienko, Gennady Andrienko, Enrique Casado, David Scarlatti, Pablo Costas, and Samet Ayhan. DART: A machine-learning approach to trajectory prediction and demand-capacity balancing. *SESAR Innovation Days*, (November), 2017. ISSN 07701268.

[16] Rui Fu, Zuo Zhang, and Li Li. Using LSTM and GRU Neural Network Methods for Traffic Flow Prediction. In *31st Youth Academic Annual Conference of Chinese Association of Automation*, pages 5–9. Institute of Electrical and Electronics Engineers Inc., 2016.

[17] Andrés Muñoz Hernández, Enrique J.Casado Magaña, and Antonio Gracia Berna. Data-driven aircraft trajectory predictions using ensemble meta-estimators. In *AIAA/IEEE Digital Avionics Systems Conference - Proceedings*, volume 2018-Septe. Institute of Electrical and Electronics Engineers Inc., 12 2018. ISBN 9781538641125. doi: 10.1109/DASC.2018.8569535.

[18] ICAO. Global TBO Concept. Technical report, ICAO Air Traffic Management Requirements and Performance Panel.

[19] John Kaneshige, Jose Benavides, Shivanjli Sharma, Lynne Martin, Ramesh Panda, and Mieczyslaw Steglinski. Implementation of a trajectory prediction function for trajectory based operations. In *AIAA AVIATION 2014 -AIAA Atmospheric Flight Mechanics Conference*, 2014. ISBN 9781624102943. doi: 10.2514/6.2014-2198. URL `http://arc.aiaa.org`.

[20] Parimal Kopardehr and Sherri Magyarits. DYNAMIC DENSITY : The Need for Dynamic Density Research Partners. pages 1–9.

[21] Jimmy Krozel and Dominick Andrisani. Intent inference and strategic path prediction. In *Collection of Technical Papers - AIAA Guidance, Navigation, and Control Conference*, volume 8, pages 6002–6017, 2005. ISBN 1563477378. doi: 10.2514/6.2005-6450. URL `http://arc.aiaa.org`.

[22] James K Kuchar and Lee C Yang. A Review of Conflict Detection and Resolution Modeling Methods. Technical Report 4, 2000.

[23] I V Laudeman, S G Shelden, R Branstrom, and C L Brasil. Dynamic density: An air traffic management metric. Technical Report April 1998, 1998. URL `https://ntrs.nasa.gov/search.jsp?R=19980210764http://tinyurl.com/aqslsy9`.

[24] Keumjin Lee, Eric Feron, and Amy Pritchett. Air traffic complexity: an input-output appraoch. In *American Control Conference*, 2007.

[25] Xiang Li, Ling Peng, Xiaojing Yao, Shaolong Cui, Yuan Hu, Chengzeng You, and Tianhe Chi. Long short-term memory neural network for air pollutant concentration predictions: Method development and evaluation. *Environmental Pollution*, 231(December):997–1004, 2017. ISSN 18736424. doi: 10.1016/j.envpol.2017.08.114.

[26] Yulin Liu and Mark Hansen. Predicting Aircraft Trajectories: A Deep Generative Convolutional Recurrent Neural Networks Approach. pages 1–24, 2018. URL `http://arxiv.org/abs/1812.11670`.

[27] Yulin Liu, Mark Hansen, David J Lovell, Michael O Ball, and Robert H Smith. Predicting Aircraft Trajectory Choice-A Nominal Route Approach. Technical report, 2018.

[28] Yulin Liu, Mark Hansen, David J Lovell, Michael O Ball, and Robert H Smith. Predicting Aircraft Trajectory Choice-A Nominal Route Approach. 2018.

[29] Go Nam LUI, Thierry Klein, and Rhea P. Liem. Data-Driven Approach for Aircraft Arrival Sequencing Investigation at Terminal Maneuvering Area. 2020. doi: 10.2514/6.2020-2869. URL `http://arc.aiaa.org`.

[30] Ioannis Lymperopoulos and John Lygeros. Adaptive aircraft trajectory prediction using particle filters. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2008. ISBN 9781563479458. doi: 10.2514/6.2008-7387. URL `http://arc.aiaa.org`.

[31] Rodrigo Marcos, Oliva G Cantú Ros, and Ricardo Herranz. Combining visual analytics and machine learning for route choice prediction: Application to pre-tactical traffic forecast. In *SESAR Innovation Days*, 2017.

[32] Anthony J Masalonis, Michael B Callaham, and Craig R Wanke. Dynamic density and complexity metrics for realtime traffic flow management. *Proceedings of the 5th USA/Europe Air Traffic Management R & D Seminar*, pages 1–10, 2003. URL `http://www.mitre.org/work/tech_papers/tech_papers_03/masalonis_tfm/masalonis_tfm.pdf`.

[33] Christopher Olah. Understanding LSTM Networks, 2015. URL `https://colah.github.io/posts/2015-08-Understanding-LSTMs/`.

[34] Patrick Pakyan Choi Martial Hebert. Learning and Predicting Moving Object Trajectory: a piecewise trajectory segment approach. Technical report, 2006.

[35] Luigi Piroddi and Maria Prandini. A geometric approach to air traffic complexity evaluation for strategic trajectory management. *Proceedings of the IEEE Conference on Decision and Control*, (October 2014): 2075–2080, 2010. ISSN 01912216. doi: 10.1109/CDC.2010.5718055.

[36] Maria Prandini and Jianghai Hu. A probabilistic approach to air traffic complexity evaluation. *Proceedings of the IEEE Conference on Decision and Control*, (March):5207–5212, 2009. ISSN 01912216. doi: 10.1109/CDC.2009.5400469.

[37] Maria Prandini, Luigi Piroddi, Stephane Puechmorel, and Silvie Luisa Brázdilová. Toward air traffic complexity assessment in new generation air traffic management systems. *IEEE Transactions on Intelligent Transportation Systems*, 12(3):809–818, 2011. ISSN 15249050. doi: 10.1109/TITS.2011.2113175.

[38] Tomislav Radisic, Doris Novak, and Biljana Juricic. Reduction of air traffic complexity using trajectory-based operations and validation of novel complexity indicators. *IEEE Transactions on Intelligent Transportation Systems*, 18(11):3038–3048, 2017. ISSN 15249050. doi: 10.1109/TITS.2017.2666087.

[39] Jasenka Rakas, Michael Seelhorst, Bona Bernard Niu, Jeffrey Tom, and Confesor Santiago. Analysis of Air Traffic Control Command Entries and the Impact on Decision Support Tool Performance. *Air Traffic Control Quarterly*, 22(1):1–20, 1 2014. ISSN 1064-3818. doi: 10.2514/atcq.22.1.1.

[40] Julia Rudnyk, Joost Ellerbroek, and Jacco M. Hoekstra. Trajectory Prediction Sensitivity Analysis Using Monte Carlo Simulations Based on Inputs' Distributions. *Journal of Air Transportation*, 27(4):181–198, 2019. ISSN 2380-9450. doi: 10.2514/1.d0156.

[41] M. Mulder S. M. B. Abdul-Rahman, C. Borst and M. M. Van Paassen. Measuring Sector Complexity: Solution Space Based Method. *Advances in Air Navigation Services*, (June 2015):11–34, 2012. doi: 10.5772/48679.

[42] M.K. Sakyi-Gyinae. A Machine Learning Approach to Evaluating Aircraft Deviations from Planned Routes. 2019.

[43] Erwan Salaün, Maxime Gariel, Adan E Vela, and Eric Feron. Aircraft proximity maps based on data-driven flow modeling. *Journal of Guidance, Control, and Dynamics*, 35(2):563–577, 2012. ISSN 07315090. doi: 10.2514/1.53859. URL `http://arc.aiaa.org`.

[44] Zhiyuan Shi, Min Xu, Quan Pan, Bing Yan, and Haimin Zhang. LSTM-based Flight Trajectory Prediction. *Proceedings of the International Joint Conference on Neural Networks*, 2018-July, 2018. doi: 10.1109/IJCNN.2018.8489734.

[45] Banavar Sridhar, Kapil S Sheth, and Shon Grabbe. Airspace Complexity and its Application in Air Traffic Management. *System*, (December):1–9, 1998. URL `http://tinyurl.com/ct5luk2`.

[46] Bence Számel, István Mudra, and Géza Szabó. Applying Airspace Capacity Estimation Models to the Airspace of Hungary. doi: 10.3311/PPtr.7512.

[47] Andrija Vidosavljevic. COTTON Final workshop - WP 2: Complexity assessment in TBO, 2019.

[48] Hongyong Wang, Ziqi Song, and Ruiying Wen. Modeling air traffic situation complexity with a dynamic weighted network approach. *Journal of Advanced Transportation*, 2018, 2018. ISSN 20423195. doi: 10.1155/2018/5254289. URL `https://doi.org/10.1155/2018/5254289`.

[49] Zhengyi Wang, Man Liang, and Daniel Delahaye. Short-term 4D trajectory prediction using machine learning methods. In *SESAR Innovation Days*, 2017.

[50] Zhengyi Wang, Man Liang, and Daniel Delahaye. A hybrid machine learning model for short-term estimated time of arrival prediction in terminal manoeuvring area. *Transportation Research Part C: Emerging Technologies*, 95(January):280–294, 2018. ISSN 0968090X. doi: 10.1016/j.trc.2018.07.019. URL `https://doi.org/10.1016/j.trc.2018.07.019`.

[51] Zhengyi Wang, Man Liang, and Daniel Delahaye. Automated Data-Driven Prediction on Aircraft Estimated Time of Arrival. *Eighth SESAR Innovation Days*, 8, 2018.

[52] B W Yap and C H Sim. Journal of Statistical Computation and Simulation Comparisons of various types of normality tests Comparisons of various types of normality tests. *Journal of Statistical Computation and Simulation*, 81(12):2141–2155, 2011. ISSN 1563-5163. doi: 10.1080/00949655.2010.520163. URL `https://www.tandfonline.com/action/journalInformation?journalCode=gscs20`.

[53] Javier Lovera Yepes, Inseok Hwang, and Mario Rotea. New algorithms for aircraft intent inference and trajectory prediction. *Journal of Guidance, Control, and Dynamics*, 30(2):370–382, 2007. ISSN 07315090. doi: 10.2514/1.26750. URL `http://arc.aiaa.org`.

[54] Junfeng Zhang, Jie Liu, Rong Hu, and Haibo Zhu. Online four dimensional trajectory prediction method based on aircraft intent updating. *Aerospace Science and Technology*, 77:774–787, 6 2018. ISSN 12709638. doi: 10.1016/j.ast.2018.03.037. URL `www.elsevier.com/locate/aescte`.

[55] Ziyu Zhao, Weili Zeng, Zhibin Quan, Mengfei Chen, and Zhao Yang. Aircraft trajectory prediction using deep long short-term memory networks. *CICTP 2019: Transportation in China - Connecting the World - Proceedings of the 19th COTA International Conference of Transportation Professionals*, (January):124–135, 2019. doi: 10.1061/9780784482292.012.