Transportation Research Part C 73 (2016) 167–182

Contents lists available at ScienceDirect

Transportation Research Part C

journal homepage: www.elsevier.com/locate/trc

Prediction of aircraft performances based on data collected by air traffic control centers

Marko Hrastovec [a,*], Franc Solina [b]

[a] Slovenia Control Ltd., Zgornji Brnik 130n, SI-4210 Brnik-Aerodrom, Slovenia
[b] Faculty of Computer and Information Science, University of Ljubljana, Večna pot 113, SI-1000 Ljubljana, Slovenia

ARTICLE INFO

Article history:
Received 19 March 2016
Received in revised form 26 October 2016
Accepted 27 October 2016

Keywords:
Aircraft performances
Trajectory calculation
Prediction
Machine learning

ABSTRACT

Accurate prediction of aircraft position is becoming more and more important for the future of air traffic. Currently, the lack of information about flights prevents us to fulfill future demands for the needed accuracy in 4D trajectory prediction. Until we get the necessary information from aircraft and until new more accurate methods are implemented and used, we propose an alternative method for predicting aircraft performances using machine learning from historical data about past flights collected in a multidimensional database. In that way, we can improve existing applications by providing them better inputs for their trajectory calculations. Our method uses flight plan data to predict performance values, which are suited individually for each flight. The results show that based on recorded past aircraft performances and related flight data we can effectively predict performances for future flights based on how similar flights behaved in the past.

1. Introduction

With the increase of air traffic, an accurate information about present and future aircraft position in airspace is becoming more and more important. Plans for future air transportation in Europe (Single European Sky ATM Research (SESAR)) and North America (NextGen) are expected to improve safety and efficiency, and match the predicted increase in air traffic. The strategic goal, envisaged for 15–20 years hence, is a new Air Traffic Management (ATM) paradigm (Brooker, 2012a,b). This new paradigm assumes that aircraft using advanced Flight Management Systems (FMS) would fly along planned 4D trajectories, incorporating altitude, position, and time. All these 4D trajectory data will be downlinked or shared. In this way, ground systems will have reliable information about trajectories and means to resolve potential conflicts (Ruiz et al., 2014). At present, downlinked data from the aircraft are not available for air traffic control or other ground systems. We believe that implementation of such pivotal changes will require a lot of time, money and resources. Before these advanced solutions become available, there is still room to improve the present operations.

To calculate 4D trajectories and predict aircraft positions we need some source of aircraft capabilities. Currently, the main source of aircraft performances is the Base of Aircraft Data (BADA) model developed by the European Organisation for the Safety of Air Navigation (EuroControl) Experimental Centre (EEC) (Eurocontrol, 2015b). In the BADA model aircraft are grouped according to their general, operational, performance, configuration and speed characteristics.

The great majority of applications for air traffic control and simulations use the BADA model to calculate performances, trajectories, times over significant points, etc. The main problem as we see it, is that these applications do not have enough

* Corresponding author.
E-mail addresses: marko.hrastovec@sloveniacontrol.si (M. Hrastovec), franc.solina@fri.uni-lj.si (F. Solina).

http://dx.doi.org/10.1016/j.trc.2016.10.018

information about individual flights. Due to this lack of specific information, trajectory calculations are in general performed with nominal input values of aircraft capabilities.

We propose to use machine learning to predict aircraft behavior that will take into account also the attributes about departure, destination, operator and other data from the flight plan that are currently available to air traffic control and are not linked directly to aircraft performances. However, these attributes can indirectly indicate particular flight performances. A distant destination, for instance, may indicate more fuel and a higher take-off weight in comparison to a close destination. On the other hand, if the fuel is expensive on the departing aerodrome, the aircraft might fuel as little as necessary. There are a lot of factors influencing the decisions that have an impact on flight performances. We cannot analyze or predict all of them, but we can discover patterns connected with attributes from flight plans, which are currently the only detailed information about flights available to air navigation service providers. With the approach that we propose we have the means to generate custom input values for trajectory calculations. Our method can predict different input values for each flight, while the nominal BADA values are carefully defined values that fit to an average flight. However, when a model is designed to fit the average, it does not fit anyone as pointed out in the book *The End of Average* by Todd Rose (Rose, 2016). Therefore, our plan is to find a fit for each flight individually and try to get in this way better predictions.

Our proposed prediction method is based on storing and analyzing real recorded flight data. We assume that flights with similar flight plan attributes would also fly similarly. Therefore, when a new flight comes, we look for similar flights from the past and predict the trajectory for the new flight based on stored data of similar flights in the past.

In our tests, we calculated flight trajectories with input values determined by our method and also with nominal values from the BADA Airline Procedures Model and Performance Table Model as the current standard. The methods of calculating trajectories were identical in both cases, only the input parameters of flight's performances and aircraft characteristics (e.g. take-off weight) were different. We have compared the predicted times with the real flight times. The results show that time predictions based on inputs from machine learning are in most cases better than predictions using nominal values.

The rest of the article is structured in four sections. In Section 2, we present the current status of aircraft performances models and some other methods to predict flight times. Our main contribution follows in Section 3, where we describe our alternative approach to the problem. In Section 4, we present the results of experiments, where we compare different approaches. Finally, in Section 5, we conclude and propose directions for future work.

2. The current situation and related work

We see the efforts in trajectory prediction going in two main directions—improving aircraft modeling and machine learning predictions based on radar inputs.

Aircraft performances models. The main source for aircraft performances is currently the BADA model (Eurocontrol, 2015b). To the best of our knowledge, there are no comparable alternatives. Currently, BADA version 3 is the most widely used version. According to Eurocontrol (Eurocontrol, 2016b), BADA is given to Air Navigation Service Providers, research and development organizations, universities, and commercial entities, to the extent necessary to enable them to work on Air Traffic Management related projects. The developers of the BADA model have identified the need for better accuracy for the new SESAR systems. In 2005 the development of BADA version 4 has started. The new model provides better accuracy and lays the ground for accurate 4D trajectory calculations needed in future systems. For now, access to BADA 4 is much more strict (Eurocontrol, 2016a) requiring a signed hard copy license, no organization-wide or multi-purpose license, proof to be able safeguard confidentiality and usage only for permitted use.

The BADA model is based on mass-varying, kinetic approach that models an aircraft as a point and requires modeling of underlying forces that cause aircraft motion (Nuic et al., 2010). BADA uses differential equations describing forces acting on the aircraft. With exact input values the trajectory is very accurate. However, the accurate input values are not available to the ground systems. The BADA user manual (Eurocontrol, 2015b) recommends improving conformance with real operations by modifying BADA default values defined in the AiRline Procedure Model (ARPM). We believe that our approach using machine learning can serve as an alternative source of input values for trajectory calculations using BADA.

Schuster et al. (2010) are using flight intent to make flight path more realistic with simulating a flight management system. Gillet et al. (2010) are using radar recordings to fine tune the AiRline Procedure Model (ARPM) which provides performance input values for the Total Energy Model. Different energy share factors and speed profiles are calculated according to airline operator, operating airport, aircraft type, flight phase and flight range with the help of statistical processing based on recorded radar data. In this way, they can generate more realistic flight trajectories for simulation purposes. They have calculated energy share factors and speed profiles just once, based on historical data, while we do the prediction dynamically for every new flight.

In 2013 an open source project called BlueSky (Hoekstra, 2016) started under the guidance of Jacco Hoekstra and based on experience with a simulation tool called the Traffic Manager (Bussink et al., 2005). This project can use the BADA model but it can also use its own aircraft model for air traffic simulations for users without a BADA licence. Metz (2015) extended BlueSky with an internal aircraft performance model which is based on publicly accessible information about aircraft. This guarantees that BlueSky remains an open source project. The internal flight dynamics model is structured similarly to the BADA model and the same algorithms are used within BlueSky for both.

Tang et al. (2015) demonstrate a method where weather forecasts are improved with Aircraft Meteorological DAta Relay (AMDAR) data and then more accurate flight profiles can be extracted from recorded flight profiles. Authors also get the trajectory points from AMDAR data and calculate the speeds from the acquired points. When they apply corrected weather data, they get more accurate aircraft performances data. Similarly, we are using an alternative path for meteorological data measured on aircraft via Mode-S transmission presented in Strajnar (2012), Strajnar et al. (2015), Hrastovec and Solina (2013).

Vilardaga and Prats (2015) present a method to plan suboptimal aircraft trajectories that have to meet time requirements at specific navigation points. The time requirements enable aircraft separation, but increase fuel burn. With strategic 4D trajectory planning the optimized trajectory plans could be produced which would minimize delays, reduce fuel burn and avoid the need for separation maneuvers at a tactical level. For strategic planning on such a scale, accurate aircraft performances for each flight are needed in advance but only predictions are available at that time.

Machine learning. De Leege et al. (2013) are using machine learning methods to predict trajectories along one particular, 45 nautical miles long, landing procedure. The trajectory prediction is calculating times over points, starting from the first approach navigation point following along significant points till the runway threshold (a total of 7 points). Model inputs are: aircraft type (heavy, medium), aircraft ground speed, altitude over the initial point and winds. The model predicts with an approximately 5 s error on the last 15 nautical miles and a 20 s error on the last 45 nautical miles trajectory. The model is optimized to predict flight times only for one approach procedure.

Kun and Wei (2008) are using radar data recordings to gather flight times. The first phase predicts the total flying time, based on recorded data of identical flights. In the second phase, the trajectory is adjusted with real-time radar data after the flight takes off. The method works only for the portion of the path that is visible to the radar.

Fablec and Alliot (1999) present a method for predicting aircraft's vertical movements with neural networks. The neural network was trained with 142 trajectories. Predictions were then performed on 50 non-learned trajectories. First, they predict a climb or descent time by knowing the aircraft type, starting altitude and final (requested) altitude. At aircraft take-off the algorithm adjusts the prediction with real flight data. The results show that neural networks are more efficient than existing non-parametric methods and that they outperform techniques used in operational systems.

Alligier et al. (2013, 2014) are dealing with the problem when exact data for reliable ground trajectory calculations are not available. The authors are estimating mass and thrust which are crucial in trajectory prediction of a climbing aircraft. The aircraft mass is estimated from a few points of the past trajectory measured with radars and the thrust law is calculated with machine learning from a training set of trajectory records. Using these input data, the computed trajectory is better than BADA model-based calculation.

Tastambekov et al. (2014) have taken an interesting approach for a mid-term conflict detection tool. The described method searches for similar trajectories from the past in terms of shape and time with k-nearest neighbors algorithm. Then a linear functional regression model is used on the already flown part of the trajectory and similar trajectories to predict the remainder of the flight.

Weitz (2013) was investigating uncertainties that influence trajectory predictions. The most influential parameters for the uncertainty are: wind, temperature, aircraft mass, speed and navigation performance. The article proves that accurate meteorological conditions and knowledge about aircraft enable the calculation of more realistic trajectories.

Kuhn (2016) uses machine learning with cluster analysis for selecting similar days at the airport to evaluate relative success of different courses of action. In this way, users can compare similar days and analyze actions to optimize future operation in similar conditions. The interesting fact is that the approach described in the paper is using several attributes such as weather conditions and air traffic flow management initiatives. In a similar vein, our method uses flight plan attributes to identify similar flights.

This article builds upon the ideas presented in two conference articles by Hrastovec and Solina (2014a,b) but adds the prediction for ARPM with more advanced methods of prediction and presents results of extensive tests.

3. Alternative approach using machine learning

We propose an alternative approach for aircraft trajectory calculation where machine learning provides BADA input parameters for trajectories calculation and makes them closer to the real trajectories instead of using BADA default or nominal input parameters. When, in the future, aircraft data including planned trajectories will be downlinked, ground systems will be able to use that data. Until then we have to find other data sources. However, for flight planning and airspace optimization, which takes place far in advance, the downlinked aircraft data will not be available even in the future. For that kind of trajectory calculations some kind of aircraft performances prediction will probably be always needed.

We propose two ways to predict input values for the calculations.

First, machine learning can predict aircraft performances in a format identical to the BADA performance tables. Many existing ATM applications, which use performance tables could read these dynamic tables instead of the static ones provided by BADA. Instead of reading a static table, a web service could be called which would provide table values predicted according to the flight plan.

The second way is to predict aircraft mass and default speeds in identical form as aircraft performance values are provided by BADA ARPM. Again, only the method for acquiring these values would be changed while all other trajectory calculation functions could remain intact.

In practice, replacing trajectory calculation normally involves the replacement of the entire ATM (sub)-system and this is the reason why some old trajectory calculation methods will probably still be in use for a long time. We propose to improve existing calculations by providing dynamic input values suited exactly for the predicted flight based on historical data of similar flights stored in a database. In that way, we can make the transition to better trajectory prediction simply by improving the accuracy of existing calculation methods.

Our method returns the input values in identical form as the BADA model does. The difference is that our values are provided via a web service and are predicted individually for every single flight. The BADA model provides low, nominal and high values as three references representing most typical flights known to BADA developers. The applications used in air traffic control usually do not have any information which custom calibrated values should be used, resulting in using the nominal average value most of the time.

3.1. Overview of the proposed method

The whole process of data acquisition, preprocessing and prediction is outlined in Fig. 1. We collect data that we use for predictions from three sources, which are available to air traffic control centers: recorded flight tracks, flight plans and meteorological data. The data sources are described in detail in Section 3.2.

All three data sources are integrated in a pre-processing phase before they are entered into the database. Flight tracks provide actual trajectories from recorded aircraft positions made by radars. We extract the actual aircraft performances for each flight from these recordings.

During these performance extractions meteorological data are also used. Wind and temperature data help us to extract accurate performances as trajectories are influenced by weather conditions. The detailed process is described in Section 3.2.3.

After we extract the aircraft performances from radar and meteorological data, they are enriched with the flight plan data. Storing the enriched performances into databases concludes the pre-processing phase. Our proposed prediction method uses the databases to predict performances for each new flight.

3.2. Data sources

3.2.1. Flight tracks

Radars are measuring positions of aircraft on every turn of the radar. Typically, the turn rates of radars are between 4 s and 12 s. Since we have normally more radars covering the airspace we get a new aircraft position practically every second. However, the radar's accuracy drops with distance and radars are not synchronized. Such disorganized and unsynchronized feeds of radar data with variable accuracy are problematic for air traffic control. To overcome these problems tracker software is used. This software receives raw radar plots and effectively minimizes radar measurement error to generate smoothed flight tracks (Farina and Pardini, 1980). Trackers combine data from multiple radars and calculate projected positions of aircraft by taking into account radar accuracies, flight capabilities, etc. Usually, some kind of filters such as Kalman filter or particle filters are used to eliminate errors and project most probable aircraft positions (Blom and Bloem, 2003; Bar Shalom et al., 1989). These smoothed tracks generated by trackers are more accurate than individual radar measurements and are the source for our calculation of aircraft performances.

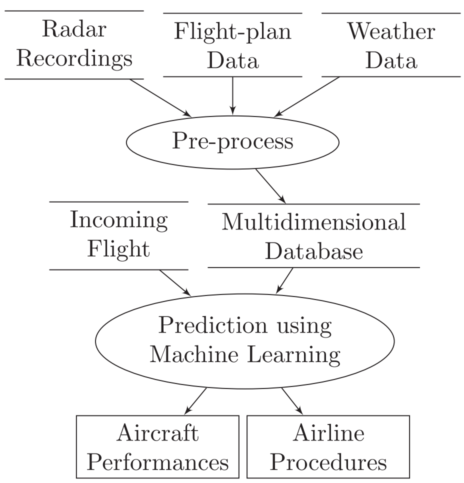


Fig. 1. The process of predicting BADA values is based on flight tracks derived from radar recordings, flight plans and weather data.

Our pre-processing procedure takes smoothed flight tracks and identifies flight phases such as level flight, climb and descent. When these flight phases are identified the performances can be extracted from each phase.

3.2.2. Flight plans

Aircraft performances and other parameters acquired from trackers are useless alone. We need an additional data source to correlate the flight plan data with a particular track and enrich raw performances with additional attributes. Flight Data Processing Systems (FDPS) in air traffic control centers handle flight plans. They use flight plans and other data to calculate when and where the flight will enter the airspace, how will it fly, where and when it will exit, etc. The flight plan is submitted by the operator to all air traffic centers where the aircraft is planned to fly. The data in flight plans are: aircraft type, operator, planned route with altitudes and speeds, aircraft equipment, date and time of the flight, duration of the flight, etc.

3.2.3. Meteorological data

The third source of data, meteorological data, is very important for extraction of correct performances from trajectories. Radars measure aircraft positions relative to radar's fixed position. Therefore, we can only measure ground speeds with radars. However, aircraft are using air speed, which is a result of a vector subtraction between ground speed and wind speed ($\overrightarrow{airspeed} = \overrightarrow{groundspeed} - \overrightarrow{windspeed}$). Therefore, wind speed is an important parameter contained in meteorological data. Temperature is another important meteorological measurement that we use. Temperature influences air density and air density has a direct influence on the climb rates of aircraft. BADA performance tables, for instance, are provided for five temperatures: ISA−20 °C, ISA−10 °C, ISA, ISA+10 °C and ISA+20 °C; where ISA is the standard ICAO atmosphere (ICAO, 1993). When calculating performances from the trajectory temperature is therefore an important parameter.

We use two sources of meteorological data. The first source of meteorological data are measurements acquired from aircraft directly with the help of Mode-S radars. Aircraft can measure temperature with their temperature sensors and wind speeds by comparing their ground speeds and air speeds. Studies from Strajnar (2012) and Hrastovec and Solina (2013) show that data acquired from aircraft are accurate enough to be used for this purpose. In Slovenia, we are routinely providing aircraft derived data to the national meteorological agency where these measurements are contributing to better weather forecasts (Strajnar et al., 2015; Hrastovec et al., 2014). The second source of meteorological data are meteorological predictions for aviation calculated in numerical weather prediction models provided by the national meteorological agency. We always take direct measurements from aircraft as the primary source if they are available. Otherwise, we take numerical weather predictions for aviation.

3.3. Pre-processing

The track data are sent from the tracker in a format called *A*ll Purpose *ST*ructured *E*urocontrol Su*R*veillance *I*nformation *EX*change (ASTERIX) (Eurocontrol, 2007) in ASTERIX Category 62 messages. Within each track message there are many fields and one of them is "Mode of Movement" which tells how the aircraft is moving. With the help of ASTERIX fields, we decompose the recorded flight tracks into climb, cruise and descent phases.

In the pre-processing step we generate two sets of BADA values. The first set are input values for airline procedure model and the second type are aircraft performances tables.

3.3.1. Estimating airline procedure model values

Airline Procedure Model (ARPM) uses Total Energy Model (TEM) to calculate performances of an aircraft along the trajectory. According to the BADA User Manual Eurocontrol (2015b), the TEM equates the rate of work done by forces acting on the aircraft to the rate of increase in potential and kinetic energy:

$$(Thr - D) \cdot V_{TAS} = mg_0 \frac{dh}{dt} + mV_{TAS}\left(\frac{dV_{TAS}}{dh}\right)\left(\frac{dh}{dt}\right) \tag{1}$$

where the variables are: *Thr*

	thrust acting parallel to the aircraft velocity vector
D	aerodynamic drag
m	aircraft mass
h	geodetic altitude
g_0	gravitational acceleration
V_{TAS}	true air speed
$\frac{d}{dh}$	time derivative

As can be observed from Eq. (1), some values, like aerodynamic drag, are determined by the aircraft type. Other values, like aircraft mass and speed, are the ones that are changing and are affecting how the aircraft is flying. These are the values, that we are trying to estimate from the recorded trajectory and later predict them in the prediction phase.

TEM Eq. (1) can be rearranged for different scenarios. We need to identify the phase of flight from the recorded trajectory and use the right formula to estimate the required aircraft parameter.

With speed and throttle controlled, Eq. (1) is used to calculate the change of altitude in time, which is the rate of climb/descent expressed as *ROCD*:

$$\frac{dH_p}{dt} = ROCD = \frac{T - \Delta T}{T}\left(\frac{(Thr - D) \cdot V_{TAS}}{mg_0}\right) f(M) \tag{2}$$

where additional variables from the previous formula are:H_p

geopotential pressure altitude
T standard atmosphere temperature
ΔT difference from standard atmosphere temperature
$f(M)$ a function of Mach number

We can rearrange Eq. (2) and get the formula for mass:

$$m = \frac{T - \Delta T}{T}\left(\frac{(Thr - D) \cdot V_{TAS}}{ROCD \cdot g_0}\right) f(M) \tag{3}$$

One should be able to just input the right values into Eq. (3) to get the estimated mass of an aircraft. However, it is not as simple as it looks. The drag D in Eq. (3) is a function of mass and velocity. Therefore, we cannot calculate the mass from Eq. (3) directly. We need to use a bisection-like method to get an estimation of mass. In the iterative method we take a number of masses in regular intervals between minimum and maximum aircraft mass. Then we calculate *ROCD* for these masses with Eq. (2) and see which one is the closest to the measured *ROCD*. We repeat the iteration with smaller steps around the best mass estimation. In each step we get the mass estimation with *ROCD* that is closer to the measured *ROCD* of a given flight. When the calculated *ROCD* differs less than ϵ from the measured one, we stop and use the mass that produced the closest *ROCD*.

Beside mass, default speeds are used for trajectory calculations. Mass and default speeds form recommended speed procedures for use in BADA TEM. With different sets of these parameters we calculate different trajectories. However, BADA always provides only airline procedure model parameters for a default company which are producing trajectories with smallest average error for flights in Eurocontrol's database. With our calculation we estimate an airline procedure profile for each flight and store it in a database. We usually cannot get all values from just one flight. From some flights we can calculate only mass and maybe a low altitude speed. From other flights we get a high altitude speed and so on.

As seen in Eq. (3), temperature is needed too. Different performances measured in different meteorological conditions may in the end result in identical mass or default speed. We have in this way normalized the calculated values since they are not weather dependent. When predicting, we can use the saved flights, as the computed values were normalized with proper weather conditions. This is not the case for extracting aircraft performances, which is described in the following Section 3.3.2, where temperature needs to be stored to use the samples for predictions recorded at similar temperatures.

3.3.2. Extracting aircraft performances

Aircraft performances are another set of values extracted from measured trajectories. These values are not calculated with the help of TEM. They are just raw performances measured from each flight. After the basic phases of a flight are identified they are further analyzed. For air speed extraction, the vector subtraction from Section 3.2.3 is used.

For the climb and descent rates we divide the whole climb/descent phase into sections which are compatible with flight levels from the BADA performance tables. The BADA performance tables provide performances for discretized flight levels and we calculate rates for the same levels. For instance, we use ranges 90–109/110–129/130–149 for flight levels 100/120/140 from the BADA performances tables.

We are simplifying the performances calculations here by assuming that all movements are of constant rate for air speeds and climb/descent rates. With this simplification, the calculated performances are averages for the flight sections observed. For our purposes of accumulating average values for prediction this is adequate. For high precision calculations, the accelerations and decelerations should be taken into account, too.

Since in this case we are not using TEM formulae, we have to store the temperatures with the extracted performances. Otherwise, one could not disambiguate whether the aircraft mass or the meteorological conditions affected the measured *ROCD* or speed.

The extracted aircraft performances and ARPM estimated data are enriched with flight plan data in the same way. Every set of values is stored with its flight plan attributes for prediction purposes.

3.4. Pre-requisites

3.4.1. Databases, technology and implementation details

Multidimensional databases that we use are created with online analytical processing (OLAP) software. Queries into an OLAP database are fast and effective with the multidimensional expressions (MDX) language (Berger et al., 2002). The prediction methods are developed on top of the OLAP database because data can be obtained with MDX queries from an OLAP database quicker and easier than from a relational database. This architecture enables us to publish the predictions as a web service. In that way, a trajectory calculation method could get customized performances suited exactly for the flight being processed.

The replacement of static BADA input values with dynamic values from the prediction allows us to keep the trajectory prediction methods the same as in the existing applications. We believe that the effort to replace just the inputs for trajectories prediction in existing applications is minor in comparison to the replacement of the entire trajectory calculation method. The scenario of replacing only the source of performance tables imposes also an important constraint in our approach. The prediction must not be too complex and should not take too much time to process. Results have to be returned from the web service in a time which enables prompt trajectory calculations.

An important feature of our pre-processing is its full automation. Every day, all three data sources are pre-processed automatically and new facts are added to the databases. When machine learning predicts values for the present day the data from the previous day are already available in the database for predictions. This gives the prediction an additional advantage that the newest trends are affecting the results. The machine learning algorithms that we use should therefore not be trained on a fixed training dataset.

The data collection described in this article is running in Slovenia Control without significant interruptions since February 2011. At present (February, 2016), there are five years of aircraft performances accumulated in our database. For the covered airspace that means that we have over twelve million facts in our aircraft performances multidimensional database and around a million and a half in the ARPM database. The reason for the much larger number of facts in the performances database is that every performance with specific atmosphere conditions is enriched with flight plan data and stored as a fact. For the ARPM values, the default values are estimated for the entire flight because they are weather independent.

3.4.2. Prediction methods requirements and limitations

The nature of the data collected in our multidimensional databases leads us to methods that should provide us with the best possible predictions.

We do not have a specific set of data which is systematically checked and corrected to be declared as a sample data used for machine learning. Since we want the prediction to use the newest data, the model should learn every day. It is unrealistic to expect that somebody would review and check the results of the learning process on a regular basis. Manual review of a learning process is a time consuming task. This is an additional reason why we have not selected a machine learning method that learns in advance. We selected a method that uses the daily updated database as it is.

Many different attributes are stored with the flight performances. Some are important and some are not. We decided not to make any a priori decision which attributes shall be used in the prediction process. Without a list of important attributes, outlier detection cannot be done. A particular performance may be an outlier according to one attribute and not by another. The final tipping point for our decision not to perform outlier detection is the sheer amount of data. We expect that the influence of outliers will be minimized by using average values. We can expect that the outliers are spread uniformly so that their impact on predicted values is negligible.

The main reasons for outliers are unreliable radar captures at the limits of radar coverage and atypical aircraft behavior. When radars return wrong positions because of coverage limits, the speed may rise or drop dramatically depending on the type of error. Another example of outliers are aircraft taxiing on the runway while tracker software is recognizing their movement as flight. In that case, speeds are significantly lower. We could have deleted such obvious outliers, but then we have to face the dilemma how far do we proceed with outlier detection and deletion. We would enter the dangerous zone of tampering with the results by deleting flights that do not fit into our models. This is also one of the reasons not to delete any flights.

3.5. Prediction

With all the pre-requisites mentioned in Section 3.4, the k-nearest neighbor (k-NN) prediction method was chosen. This method is also called a "lazy" learning method (Kononenko and Kukar, 2007). We have evaluated other algorithms, but the k-NN has the features we are looking for. The burden of computation is moved from the learning to the prediction phase and that is a drawback that we had to consider since predictions take more time. The algorithm allows also accurate numerical predictions. Other machine learning algorithms mainly try to make a classification of the training samples and have problems with exact numerical values. They solve the problem of numerical prediction by discretization. Numerical prediction then usually means classifying the sample into one of the discretized values which is not the best fit any more.

As in many other cases, the k-NN algorithm could not be used directly. We had to adapt it to our specific problem. First, the distance needs to be defined in order to find the k nearest neighbors. The distance in our case cannot be an Euclidean distance. Our measure of distance is the count of non-matching attributes for two given flights:

$$d_{x,y} = \sum_{i=1}^{n}[x_i \neq y_i] \tag{4}$$

where:

$d_{x,y}$ distance for similarity measure between flights x and y

n the number of attributes

x_i, y_i i-th attributes

With a distance measure like this, a lot of pairs have identical distances. However, all attributes are not equally significant in determining aircraft performances. It is not important only how many attributes are not matching, but also which are the ones that do match. Therefore, the algorithm should take care which attributes to compare when looking for the closest neighbors.

Our task is to find the set of attributes which locate the closest neighbors in terms of aircraft performances. With a good set of k nearest neighbors we can get a good prediction. If the selection of significant attributes is not good we will not get the closest neighbors and the prediction will be poor. With these assumptions our problem of prediction reduces to the problem of finding the set of attributes that determine the set of flights exhibiting the closest aircraft performances.

Similarly as the naive Bayesian classifier does (Kononenko and Kukar, 2007), we assume independence of attributes, although this is not actually the case. For all attributes our algorithm tries to establish their significance and sorts them accordingly. The significance of the attribute should tell us how strongly the attribute is linked to flight performance. When the most significant attributes for finding the set of the k nearest neighbors will be used, the best prediction will be obtained.

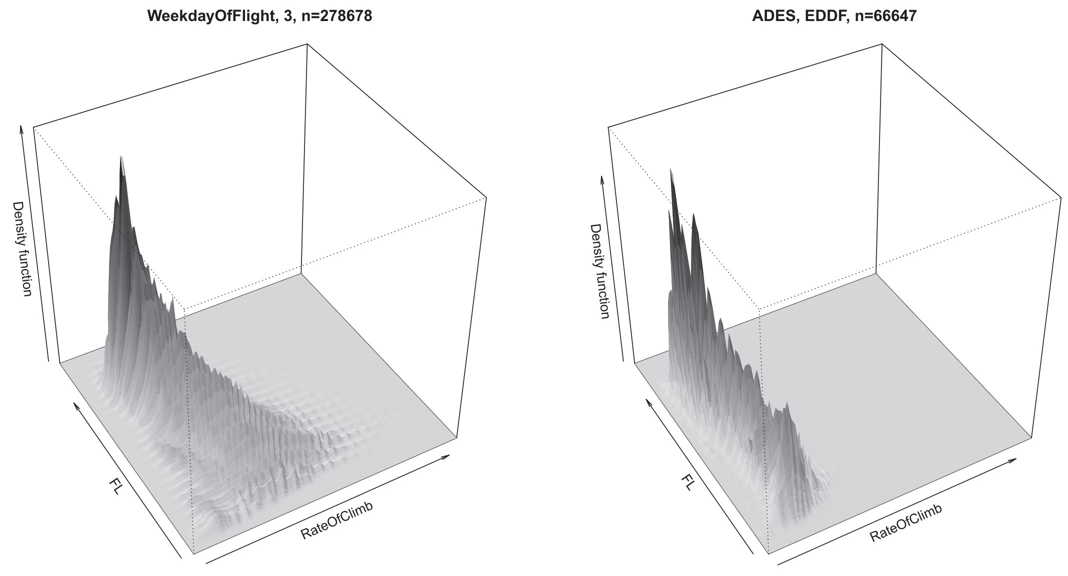
One of the approaches that we tried in our experiments was a manual selection of attributes. This can be considered as an expert guess or judgment on which attributes should provide good results. However, one can only make one static selection of attributes in advance for all the predictions. On the other hand, our machine learning approach makes a decision which attributes to use for every prediction separately. We call this dynamic attribute selection.

To evaluate and select attributes dynamically we use the spread of the values. If a particular attribute is not important for the performance of the aircraft, the values of measured performances are spread more uniformly. On the other hand, if an attribute is important, the values should be concentrated more closely around the actual performance.

For example, Fig. 2a shows, how the attribute "weekday of flight" (value 3 – Tuesday) is affecting the rate of climb. It can be seen that the "weekday of flight" attribute probably does not influence the rate of climb significantly because the density function values are widely spread. On the other hand, Fig. 2b shows more concentrated values of the rate of climb for the attribute "aerodrome of destination" (ADES) for Frankfurt (EDDF). In this case, an attribute such as "aerodrome of destination" can tell us more about the expected rate of climb than the "weekday of flight". Local maximums, which can be observed in both Fig. 2a and b belong to other attributes which have not been fixed to a value like the observed attribute was.

Since we took a naive approach we are not considering inter-dependence of attributes. If we would want to evaluate also all possible combinations of attributes, the time complexity of the prediction would rise from $\mathcal{O}(n)$ for single attributes to $\mathcal{O}(2^n)$ for all their possible combinations, where n represents the number of attributes. The latter would require too much time and resources. For illustration, we have generated around 1000 charts like Fig. 2a and b for the most frequent values of attributes to visualize how different attribute values are spread. If we would want to make similar charts only for the combination of two attributes, we would have to generate one million charts. It is computationally impossible to evaluate the performances for all combinations of attributes in every prediction.

The proposed prediction algorithm is outlined in Algorithm 1. When facts about the predicted flight are known, dispersion values for each attribute value of this flight are extracted from the database.



(a) Dispersed influence (b) Concentrated influence

Fig. 2. Comparison of the influence of attributes (a) "weekday of flight" and (b) "aerodrome of destination" on the expected rate of climb. (FL = Flight Level).

Algorithm 1. Prediction algorithm

```
function GetValues(attribute values of the flight going to take off)
  get standard deviation σ for each attribute value
  sort attributes according to their σ
  attrSet ← sorted attributes
  prediction ← ∅
  repeat
    currentValues ← sets of samples for the given attrSet
    for All Values in currentValues do
      if set of samples bigger than k members then
        calculate average value of the given set
        prediction ← calculated average
      end if
    end for
    attrSet ← attrSet − last attribute      ▹ remove the attribute with the largest σ
  until all predictions calculated or attrSet empty
  for all missing predictions do
    it exists average behavior for aircraft type in the database then
      prediction ← average value for aircraft type from the database
    else
      prediction ← nominal value from BADA
    end if
  end for
  return prediction
end function
```

Each variable (air speed, rate of climb, rate of descent, aircraft mass, default speed) gets its own set of standard variation values. After these standard variations are sorted the search for the predicted value begins. First, the whole set of attributes is used to get the flights from the database. That means that only the flights with identical set of all evaluated attributes are used. If there are not enough records with matching attributes in the database, the condition is relaxed by deleting the most dispersed attribute. This relaxation is repeated until there are at least k records in the database to calculate average values from multiple measurements. For rare flights with less frequent attribute values there may not be enough samples in the database. In that case, the algorithm tries to get the average values for the aircraft type. If even the aircraft type is new to the considered airspace, the prediction returns the nominal BADA values for the given aircraft type.

4. Results

4.1. Test data and methodology

The test set for our prediction method were flights recorded in the airspace "visible" to Slovenia Control radars for the period January to June 2015. That is approximately 115,000 flights with corresponding flight plans. We have in the database also the actual recorded flight times which serve as a reference.

The method for calculating predicted flight times is identical for all different methods of predictions. Only the input values of performances were different. The whole test procedure is outlined in Fig. 3.

First, we identify the sections of each flight for which we predict flight times. Since we have recordings for the flights, we know the actual times for each section as a reference.

For each flight from the test set and its sections we calculate how long the flight would take in a particular section if it would be flying according to the predicted performance values. The calculated times for sections, calculated with predicted performances, are different from the actual times flown. The difference between calculated times and actual times is our indicator of prediction quality.

We were considering to predict flight times also by using flight plan routes, departure and approach procedures, etc. But in such a scenario, we would not have any information about deviations from routes and controller's clearances which are a very common practice. Practically no aircraft is following the points on the route exactly as planned in the flight plan. Clearances to make shortcuts and to fly directly to distant points are a regular practice. Similarly, a lot of approach and departure procedures are done in a way to make the path shorter and to enable the operators to save fuel and time. There are also clearances to change altitude or direction in order to avoid other aircraft or bad weather conditions to fly safely through the airspace. In our test system we do not have the information which clearances or shortcuts have been given to the flights. Therefore, we cannot calculate from the flight plan how long the flight should have taken. We decided to use a method which

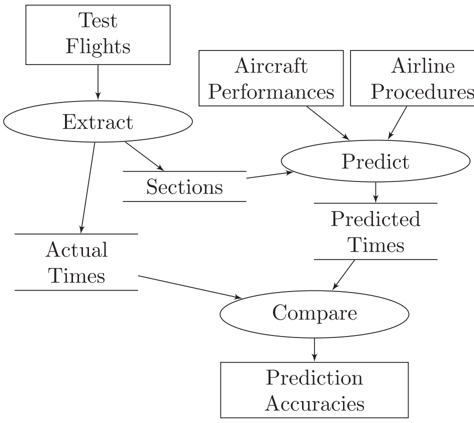


Fig. 3. The procedure of evaluating prediction accuracies.

compares directly the calculated flight times with the real ones which were recorded with radars in order to evaluate the prediction accuracy and not flight plan route calculations.

The first source of aircraft performances in our comparison study was BADA model version 3.13. The model provides three values for each performance: low, nominal and high. We used the nominal value. This source is labeled as "BADA" in the charts.

The second source of aircraft performances data were average performances calculated from our database based only on the aircraft type. No other attributes were taken into account in this case. This source is labeled as "AC Type Average".

The next source is the lazy learning method which uses a limited manually selected fixed set of attributes for searching the k nearest neighbors. The set of attributes was: aircraft type, operator, aerodrome of destination, aerodrome of departure. Apart from attribute selection, the k-NN algorithm is the same as in our method described in Section 3.5. This source is labeled as "Lazy Learning" in the charts. This kind of feature selection is called global feature selection because the features are ranked and selected once globally for all predictions.

The last source is our machine learning method with dynamic allocation of attributes described in Section 3.5. The attributes used are: aerodrome of departure/destination, aircraft type, arrival/departure hour, airspace entry/exit point, exemption (state, hospital, ...), flight rule (instrumental, visual, ...), flight type (scheduled, non-scheduled, ...), operator, weekday. This source of aircraft performances is labeled as "Dynamic Attributes" in the charts.

We used the prediction with both databases. First, the trajectories were calculated with predicted aircraft performance tables and next with ARPM values acquired with the help of TEM.

We have changed only one detail in the algorithm described in Section 3.5. The aircraft type is always the most significant attribute in Dynamic Attributes algorithm regardless of its coefficient of variation. This change was done because the aircraft type is implicitly included in the TEM formulae.

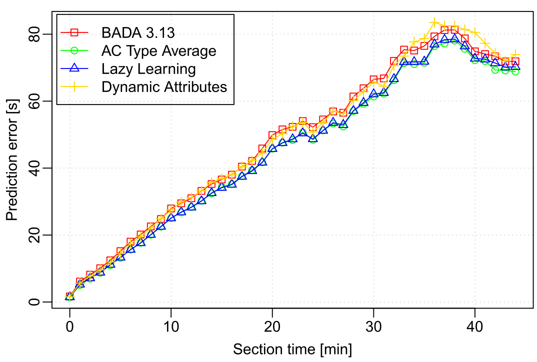


Fig. 4. Average error for air speed predictions with aircraft performances.

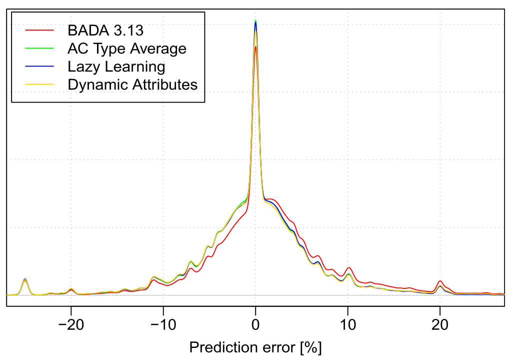


Fig. 5. Error distribution for air speed predictions with aircraft performances.

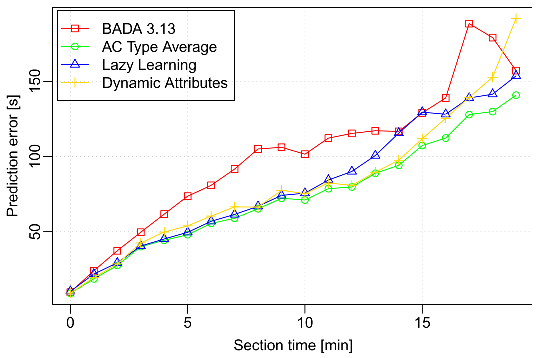


Fig. 6. Average error for climb rate predictions with aircraft performances.

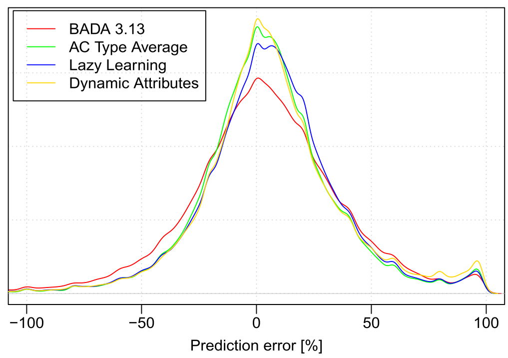


Fig. 7. Error distribution for climb rate predictions with aircraft performances.

4.2. Comparison of prediction methods

4.2.1. Aircraft performances

Figs. 4, 6 and 8 show the predictions, which use aircraft performance tables. The charts show average absolute prediction errors in seconds. Higher values mean greater errors. Lower values mean lower errors and more accurate predictions. On the *x*-axes the lengths of the flight sections in minutes are represented, showing that for longer sections the prediction errors are larger although relatively to the section length the errors are smaller with longer sections.

Fig. 4 shows the prediction accuracy of lateral movements. We have tried several prediction methods in addition to the ones shown in this article. Some of them managed to predict as good as the best two shown in Fig. 4 but none could perform better. As Fablec and Alliot (1999) say, horizontal prediction is quite accurate and there are speeds in the flight plans to follow. We can see in Fig. 4 that AC Type Average and Lazy Learning perform slightly better than the method with dynamic attributes or BADA.

Climb rates in Fig. 6 show that more sophisticated and tailored methods produce smaller errors. The AC Type Average and Dynamic Attributes algorithms show the best results in climb predictions. The Lazy Learning algorithm is slightly worse for longer sections while BADA tables produce significantly larger errors for climbs.

For the descent prediction in Fig. 8 we can see that all machine learning methods outperform BADA provided aircraft performances. In this case the Dynamic Attributes algorithm gives the best results while Lazy Learning is the worst.

Overall it seems that for all three kinds of trajectories the Dynamic Attributes algorithm or plain AC Type Average are the best candidates.

In Figs. 5, 7 and 9 the distributions of errors are presented. We can see that the majority of prediction errors are concentrated around 0% except BADA for descents in Fig. 9 which has another peak around 70%. This peak on the positive side means that predicted times were shorter than the ones actually flown. This peak represents a large number of mispredictions of descent rates on altitudes around FL360 ± 30. On this altitude the tropopause (boundary between troposphere and strato-

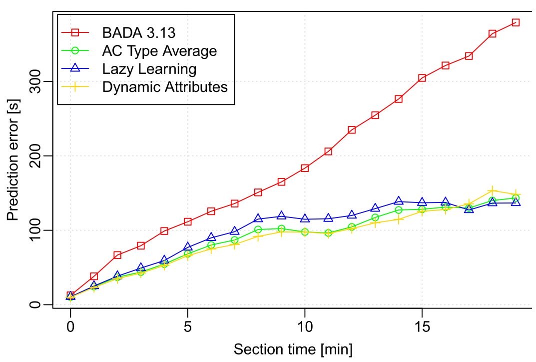


Fig. 8. Average error for descent rate predictions with aircraft performances.

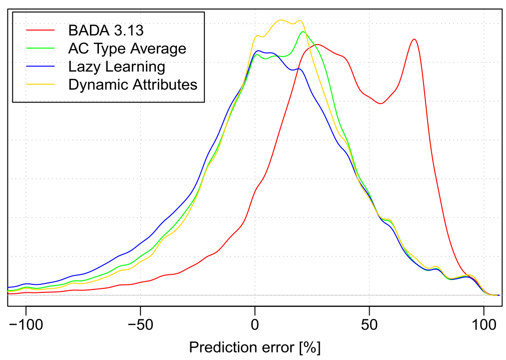


Fig. 9. Error distribution for descent rate predictions with aircraft performances.

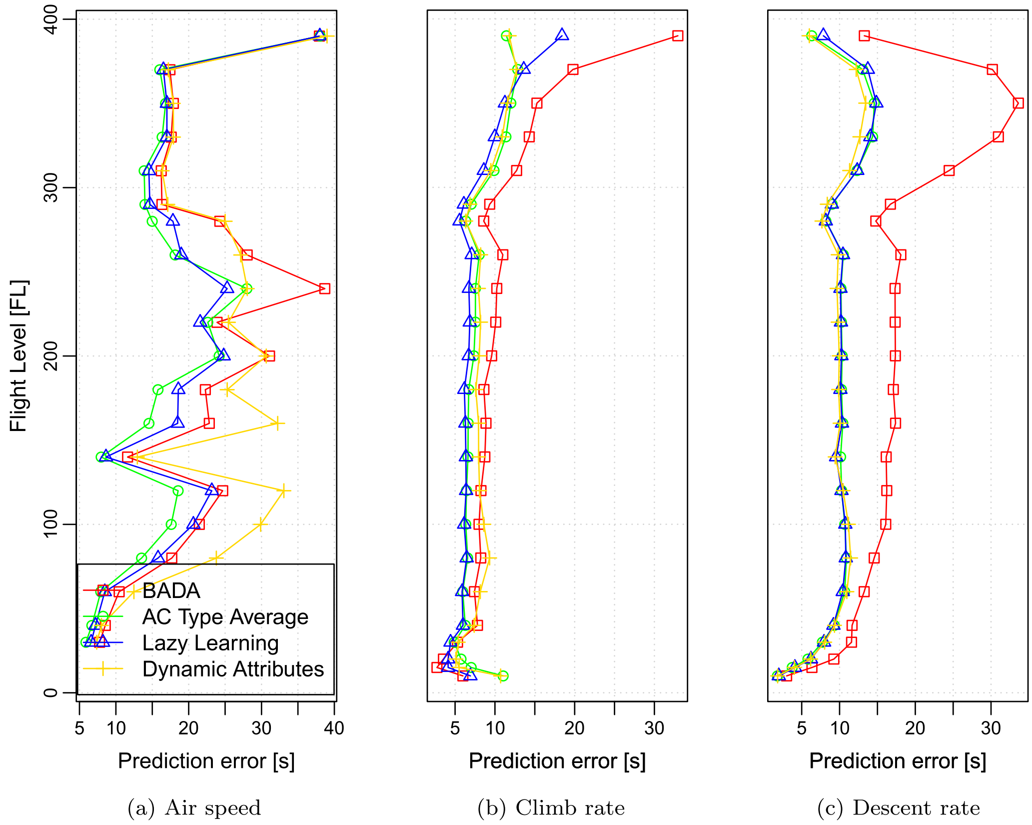


Fig. 10. Aircraft performance prediction error in relation to aircraft altitude.

sphere) is located. Different calculations are used for troposphere and stratosphere. Since the tropopause is not very sharp and fixed, wrong formulae have obviously been used. BADA inputs are the most affected here because the values were calculated using TEM while others ignore the physical model and use recorded times data from the database.

It is interesting to see how the aircraft performance predictions are affected by altitude in Fig. 10. The predictions seem to be less accurate for the air speed, but the sections where air speeds are measured and predicted are longer and therefore the errors are a bit larger in absolute. Here we can observe again that the machine learning model gives better results especially for climbs and descents. Larger errors of descent predictions through tropopause in Fig. 10c correspond to the distribution peak of BADA in Fig. 9.

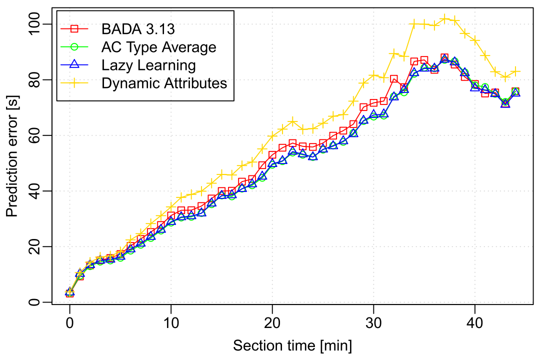


Fig. 11. Average error for air speed predictions with ARPM inputs.

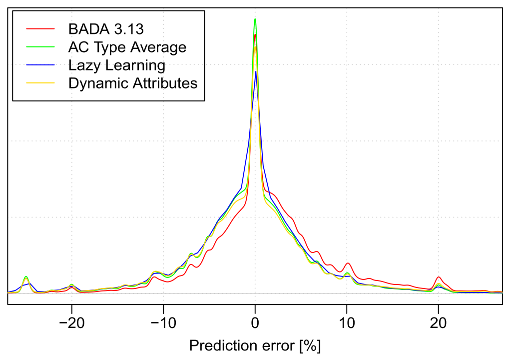


Fig. 12. Error distribution for air speed predictions with ARPM inputs.

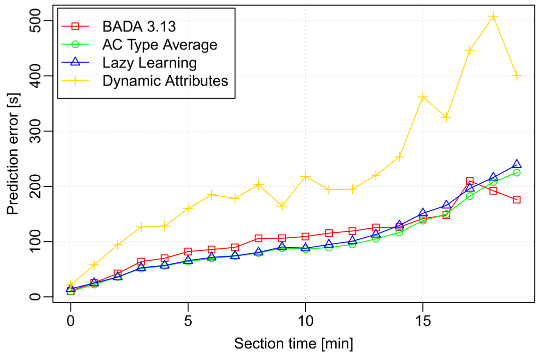


Fig. 13. Average error for climb rate predictions with ARPM inputs.

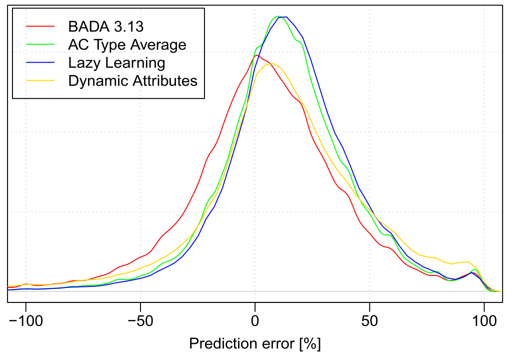


Fig. 14. Error distribution for climb rate predictions with ARPM inputs.

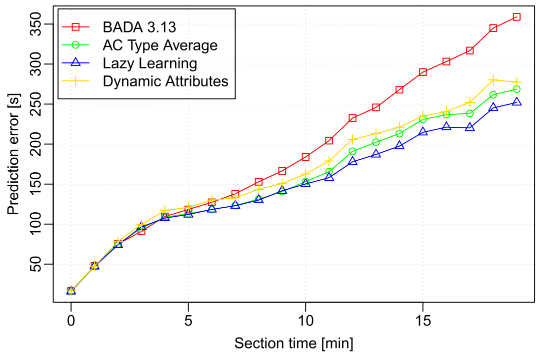


Fig. 15. Average error for descent rate predictions with ARPM inputs.

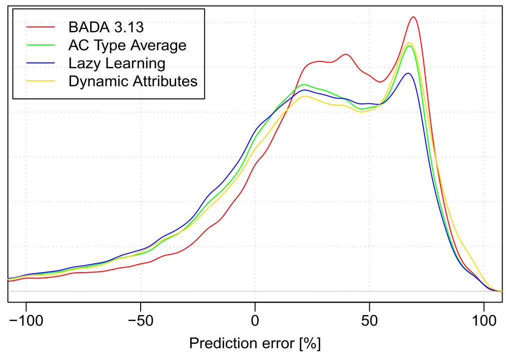


Fig. 16. Error distribution for descent rate predictions with ARPM inputs.

4.2.2. Total energy model

While we can observe a better performance of machine learning methods for aircraft performances predictions, the situation is somewhat different when calculating trajectories with custom ARPM values for TEM in Figs. 11, 13 and 15.

For lateral movements in Fig. 11 the Dynamic Attributes algorithm gives consistently worst results over all section lengths. However, the other two machine learning methods are still a bit better than BADA nominal values.

The Dynamic attributes algorithm is even worse for climbs in Fig. 13. It is obvious that this method is not suitable for climbs. The Lazy Learning and AC Type Average are again a bit better than nominal BADA.

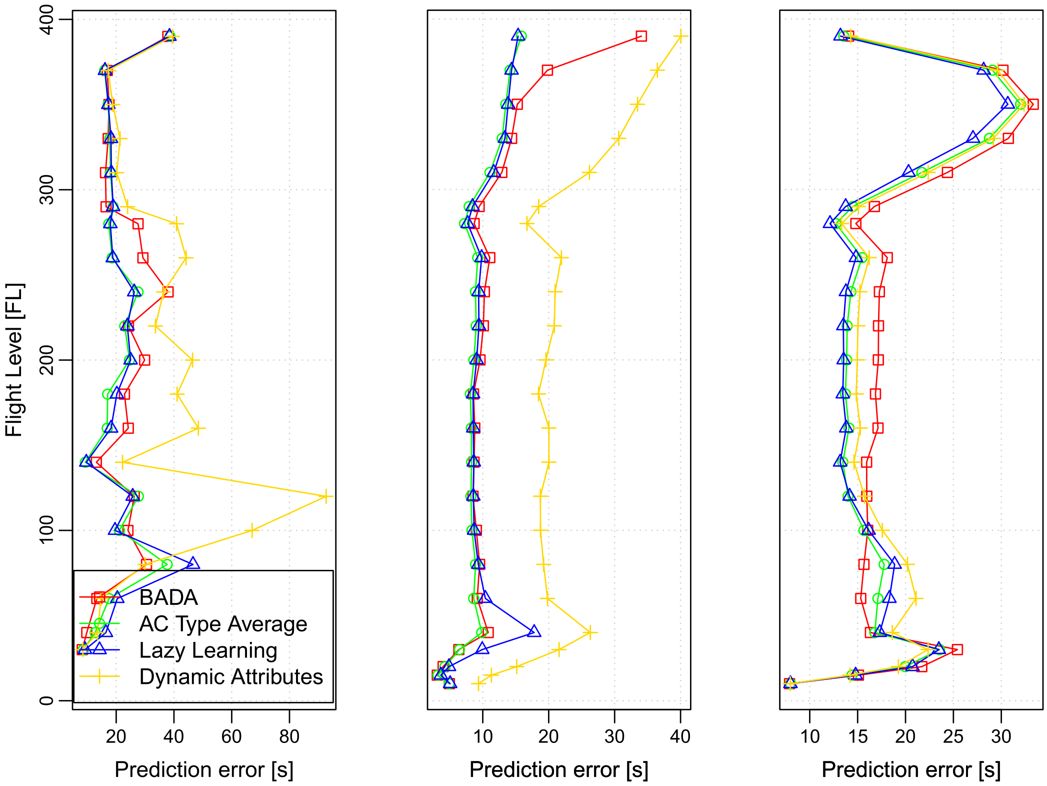
The same problem of tropopause can be observed again in Fig. 15. The Lazy Learning method is the best while Dynamic Attributes algorithm proves again that it is more suitable for aircraft performances predictions than for ARPM.

Figs. 12 and 14 show again a concentration of errors around 0%. Fig. 16 is different and shows similar peaks as BADA in Fig. 9. Unlike aircraft performances predictions, this time all calculations are affected because they all use TEM. It seems that both presented methods should be combined and aircraft performances methods should be used for tropopause altitudes.

In Fig. 17 we can see again that for the ARPM the Dynamic Attributes learning algorithm does not perform very well. Bad predictions of descents in Fig. 17c correspond to peaks in Fig. 16.

5. Conclusions

The results show that the use of recorded flight data for aircraft performance prediction is promising. The predictions made by different machine learning techniques based on recorded data are close to each other. The predictions using the BADA nominal values are giving the worst results because the model is made for a general case suitable for the whole world and does not include local characteristics. This is the reason, why ARPM is designed to customize TEM and our prediction method enables these local customizations.



(a) Air speed (b) Climb rate (c) Descent rate

Fig. 17. ARPM inputs prediction error in relation to aircraft altitude.

We have expected that the machine learning methods would provide even better results, but that was probably an unrealistic expectation. If the experts know how to evaluate attributes and how to use them to get a credible value we cannot expect that machine learning algorithms will come up with a magical recipe that will improve the predictions drastically. The main advantage of a machine learning approach in our case is to make good predictions much faster and cheaper, based on a large amount of data, and to take the burden of routine tasks off the human expert.

There is still room for improvement in the machine learning methods presented in this article. We see it in fine tuning of ensemble learning which would prioritize a prediction based on their performance on a particular section of the trajectory.

We see two potential usages of predictions. One could generate an airline procedure model data for various companies from the historical data stored in the database. The second usage would be to use the machine learning results instead of static nominal values from the BADA. The computation of predictions is fast enough to be used in real-time day to day operations. For now the service is running for test purposes.

All aircraft performances prediction algorithms are installed as a web service. Legacy applications using BADA values for trajectory calculation could use this service and get better performances prediction. All methods return performances in a form identical to BADA performance tables. In that way, they could replace the BADA nominal values in legacy applications which would calculate more accurate trajectories with minimal changes and consequently at a minimal cost.

We wanted to prove that our algorithm works well for any season of the year. We have chosen for testing on purpose a time period where calm winter season transforms into high summer season. In that way, we have covered most of the seasonal traffic patterns in our airspace.

OLAP databases support cluster architecture and large amounts of data. They are optimized to provide quick answers to queries. The multidimensional database is designed to be easily expanded to a larger airspace. In our case, we were able to perform all the tasks with a couple of desktop computers. With powerful servers and careful planning, expanding to a larger airspace should not be a problem. In that case, the predictions should be based on geographic position to support local operations characteristics. The databases are already designed to hold this information. The OLAP supports partitioning for such scenarios. Partitioning enables the distribution of the database and optimization of queries to work in parallel and to search only through the partitions which actually hold the data searched for. In that way, the predictions with larger databases would not require much more time.

A database holding data of a large airspace becomes a good candidate for a centralized service providing predictions to a wider audience. In Europe we have good experience with centralized services. Eurocontrol's Network Manager Operations Centre (NMOC), for instance, delivers core operational services across several domains in flow and capacity management, flight planning, etc. Eurocontrol is trying to introduce new centralized services (Eurocontrol, 2015a) which would provide consistent and cheaper services across the whole Europe. A proposal for a centralized service named "CS2: 4D Trajectory Flight Profile Calculation for Planning Purposes Service (4DPP)" (Eurocontrol, 2013) should bring a consistent 4D trajectory calculation for all Eurocontrol stakeholders. A service like that would use more advanced BADA TEM trajectory prediction. However, a database like ours could help in tuning the model and provide local or operator characteristics for better performance. With centralized services, the 4D trajectory calculations would be more consistent and accurate in the whole European airspace at a lower price. There would be no need for every air traffic control center to invest money in calculating them by themselves.

Acknowledgments

This research is partly supported by the European Union – the European Social Fund.

References

Alligier, R., Gianazza, D., Durand, N., 2013. Learning the aircraft mass and thrust to improve the ground-based trajectory prediction of climbing flights. Transp. Res. Part C: Emerg. Technol. 36, 45–60. http://dx.doi.org/10.1016/j.trc.2013.08.006.

Alligier, R., Gianazza, D., Ghasemi Hamed, M., Durand, N., 2014. Comparison of two ground-based mass estimation methods on real data. In: 8th International Conference on Research in Air Transportation., pp. 1–8.

Bar Shalom, Y., Chang, K.C., Blom, H.A.P., 1989. Tracking a maneuvering target using input estimation versus the interacting multiple model algorithm. IEEE Trans. Aerosp. Electron. Syst. 25 (2), 296–300.

Berger, A., Netz, A., Pasumansky, M., 2002. Record for Multidimensional Databases. US Patent 6,442,560.

Blom, H., Bloem, E., 2003. Tracking multiple maneuvering targets by joint combinations of IMM and PDA. In: 42nd IEEE International Conference on Decision and Control (IEEE Cat. No. 03CH37475) (December), pp. 2965–2970. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1273077>.

Brooker, P., 2012a. 4D-trajectory air traffic management: are there 'Killer Apps'? Part 1. J. Navigation 65 (03), 397–408.

Brooker, P., 2012b. 4D-trajectory air traffic management: are there 'Killer Apps'? Part 2. J. Navigation 65 (04), 571–587.

Bussink, F., Hoekstra, J., Heesbeen, B., 2005. Traffic manager: a flexible desktop simulation tool enabling future ATM research. 24th Digital Avionics Systems Conference, vol. 1. IEEE, pp. 3.B.4-1–3.B.4-10. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1563344>.

De Leege, A., van Paassen, M., Mulder, M., 2013. A machine learning approach to trajectory prediction. In: AIAA Guidance, Navigation, and Control (GNC) Conference. American Institute of Aeronautics and Astronautics, Reston, Virginia. URL <http://arc.aiaa.org/doi/abs/10.2514/6.2013-4782>.

Eurocontrol, November 2007. Part 1 All Purpose Structured Eurocontrol Surveillance Information Exchange (ASTERIX), first ed. URL <http://www.eurocontrol.int/asterix/gallery/content/public/documents/pt1ed130.pdf>.

Eurocontrol, 2013. Centralised Service on 4D Trajectory Calculation for Planning Purposes (4DPP) Concept of Operations (CONOPS). URL <http://www.eurocontrol.int/sites/default/files/centralised-services/cs2-4dpp-conops-comments-ed-2-0-23-10-2013.pdf>.

Eurocontrol, 2015a. Centralised Services. URL <http://www.eurocontrol.int/centralised-services>.

Eurocontrol, 2015b. User Manual for the Base of Aircraft Data (BADA) Revision 3.13. Tech. Rep. May, Eurocontrol.

Eurocontrol, 2016a. BADA 4 Family Licence Request. URL <https://badaext.eurocontrol.fr/licence/bada/v4>.

Eurocontrol, 2016b. Base of Aircraft Data (BADA). URL <https://www.eurocontrol.int/services/bada>.

Fablec, Y.L., Alliot, J.M., 1999. Using neural networks to predict aircraft trajectories. In: International Conference on Artificial Intelligence, pp. 524–529.

Farina, A., Pardini, S., 1980. Survey of radar data-processing techniques in air-traffic-control and surveillance systems. IEE Proceedings F Communications, Radar and Signal Processing 127 (3), 190. URL <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4644606>, <http://digital-library.theiet.org/content/journals/10.1049/ip-f-1.1980.0030>.

Gillet, S., Nuic, A., Mouillet, V., 2010. Enhancement in realism of ATC simulations by improving aircraft behaviour models. In: 29th Digital Avionics Systems Conference. IEEE, pp. 2.D.4-1–2.D.4-13. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5655482>.

Hoekstra, J., 2016. Bluesky. URL <http://homepage.tudelft.nl/7p97s/BlueSky/download.html>.

Hrastovec, M., Solina, F., 2013. Obtaining meteorological data from aircraft with Mode-S radars. IEEE Aerosp. Electron. Syst. Mag. 28 (12), 12–24. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6693664>.

Hrastovec, M., Solina, F., 2014a. Machine learning model for aircraft performances. In: 33rd Digital Avionics Systems Conference, pp. 8C4-1–8C4-10. URL <http://ieeexplore.ieee.org/xpl/articleDetails.jsp?tp=&arnumber=6979541>.

Hrastovec, M., Solina, F., 2014b. Machine learning model for aircraft performances. In: 6th International Conference on Research in Air Transportation. URL <http://www.icrat.org/icrat/Author/MarkoHrastovec488/FINAL-559-cfp-Hrastovec.pdf>.

Hrastovec, M., Strajnar, B., Solina, F., 2014. Transmission and usage of local airborne measurements for weather forecasting. Uporabna informatika XXII (4), 202–208.

ICAO, 1993. Doc 7488, Manual of the ICAO Standard Atmosphere: Extended to 80 kilometres (262 500 feet), third ed. International Civil Aviation Organization.

Kononenko, I., Kukar, M., 2007. Machine Learning and Data Mining. Woodhead Publishing.

Kuhn, K.D., 2016. A methodology for identifying similar days in air traffic flow management initiative planning. Transp. Res. Part C: Emerg. Technol. 69, 1–15. http://dx.doi.org/10.1016/j.trc.2016.05.014.

Kun, W., Wei, P., 2008. A 4-D trajectory prediction model based on radar data. In: 27th Chinese Control Conference. IEEE, pp. 591–594. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4605732>.

Metz, I., 2015. Aircraft Performance for Open Air Traffic Simulations. Master's Thesis.

Nuic, A., Poles, D., Mouillet, V., 2010. BADA: an advanced aircraft performance model for present and future ATM systems. Int. J. Adapt. Control Signal Process. 24 (10), 850–866.

Rose, T., 2016. The End of Average. Allen Lane.

Ruiz, S., Piera, M.a., Nosedal, J., Ranieri, A., 2014. Strategic de-confliction in the presence of a large number of 4D trajectories using a causal modeling approach. Transp. Res. Part C: Emerg. Technol. 39, 129–147. http://dx.doi.org/10.1016/j.trc.2013.12.002.

Schuster, W., Ochieng, W., Porretta, M., 2010. High-performance trajectory prediction for civil aircraft. In: 29th Digital Avionics Systems Conference.. IEEE, pp. 1.C.3-1–1.C.3-9. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5655515> .

Strajnar, B., 2012. Validation of Mode-S meteorological routine air report aircraft observations. J. Geophys. Res.: Atmosp. 117 (D23). http://dx.doi.org/10.1029/2012JD018315.
Strajnar, B., Žagar, N., Berre, L., 2015. Impact of new aircraft observations Mode-S MRAR in a mesoscale NWP model. J. Geophys. Res.: Atmosp. URL http://doi.wiley.com/10.1002/2014JD022654 .
Tang, X., Chen, P., Zhang, Y., 2015. 4D trajectory estimation based on nominal flight profile extraction and airway meteorological forecast revision. Aerosp. Sci. Technol. 45, 387–397. http://dx.doi.org/10.1016/j.ast.2015.06.001.
Tastambekov, K., Puechmorel, S., Delahaye, D., Rabut, C., 2014. Aircraft trajectory forecasting using local functional regression in Sobolev space. Transp. Res. Part C: Emerg. Technol. 39, 1–22. http://dx.doi.org/10.1016/j.trc.2013.11.013.
Vilardaga, S., Prats, X., 2015. Operating cost sensitivity to required time of arrival commands to ensure separation in optimal aircraft 4D trajectories. Transp. Res. Part C: Emerg. Technol. 61, 75–86. http://dx.doi.org/10.1016/j.trc.2015.10.014.
Weitz, P., 2013. Determination and visualization of uncertainties in 4D-trajectory prediction. In: Integrated Communications, Navigation and Surveillance Conference, ICNS, pp. 1–9.