Proceedings of the 2018 Winter Simulation Conference
M. Rabe, A.A. Juan, N. Mustafee, A. Skoogh, S. Jain, and B. Johansson, eds.

PREDICTION OF AIRCRAFT BOARDING TIME USING LSTM NETWORK

Michael Schultz
Stefan Reitmann

Institute of Flight Guidance
German Aerospace Center
Lilienthalplatz 7
Braunschweig, 38108, GERMANY

ABSTRACT

Reliable and predictable ground operations are essential for 4D aircraft trajectories. Uncertainties in the airborne phase have significantly less impact on flight punctuality than deviations in aircraft ground operations. The ground trajectory of an aircraft primarily consists of the handling processes at the stand, defined as the aircraft turnaround, which are mainly controlled by operational experts. Only the aircraft boarding, which is on the critical path of the turnaround, is driven by the passengers' experience and willingness or ability to follow the proposed procedures. We propose a machine learning approach predict the boarding time. A validated boarding simulation provides data input for a recurrent neural network approach (discrete time series of boarding progress). In particular we use a Long Short-Term Memory model to learn the characteristic passenger behaviors over time.

1 INTRODUCTION

From an air transportation system view, a flight could be seen as a gate-to-gate or an air-to-air process. Whereas the gate-to-gate is more focused on the aircraft trajectory flown, the air-to-air process concentrates on the airport ground operations to enable efficient flight operations proving reliable departure times. Typical standard deviations for airborne flights are 30 s at 20 minutes before arrival (Bronsvoort et al. 2009), but could increase to 15 min when the aircraft is still on the ground (Mueller and Chatterji 2002). To evaluate these deviations in an economic context, Cook and Tanner (2015) provide reference values for the cost of delay to European airlines. The average time variability (measured as standard deviation) is in the flight phase (5.3 min) smaller than the variability of both departure (16.6 min) and arrival (18.6 min) (Eurocontrol 2015). If the aircraft is departing from one airport, changes with regards to the arrival time at the next are comparatively small (Tielrooij et al. 2015). Current research in the field of flight operations primarily addresses economic and ecological efficiency (Rosenow et al. 2017a, 2017b, 2018; Niklaß et al. 2017).

The aircraft turnaround on the ground consists of the major ground handling operations at the stand: deboarding, catering, cleaning, fueling and boarding as well as the parallel processes of unloading and loading (Airbus 2017). All these handling processes follow clearly defined procedures and are mainly controlled by ground handling, airport or airline staff (Fricke and Schultz 2008, 2009). But in particular, aircraft boarding is driven by passengers' experience and willingness or ability to follow the proposed procedures and is disturbed by individual events, such as late arrivals, no-shows, specific (high) numbers of hand luggage items, or priority passengers (privileged boarding). To provide reliable values for the target off-block time, which is used as a planning time stamp for the subsequently following departure procedures, all critical turnaround processes are subject to prediction. In this context, the complex, stochastic, and passenger-controlled boarding makes it difficult to reliably predict the turnaround time, even if boarding is already in progress.

978-1-5386-6572-5/18/$31.00 ©2018 IEEE

1.1 Status Quo

Our research is connected to three different topics: aircraft turnaround, passenger behavior, and machine learning. Comprehensive overviews are provided by Schmidt (2017) for aircraft turnaround, by Jaehn and Neumann (2015) for boarding, and by Nyquist and McFadden (2008) and Mirza (2008) for the corresponding economic impact (Cook and Tanner 2015). Relevant studies include, but are not limited to, the following current examples.

With a focus on efficient aircraft boarding, Milne and Kelly (2014) develop a method that assigns passengers to seats so that their luggage is distributed evenly throughout the cabin, assuming a less time-consuming process for finding available storage in the overhead bins. Qiang et al. (2014) propose a boarding strategy that allows passengers with a large amount of hand luggage to board first. Milne and Salari (2016) assign passengers to seats according to the number of hand luggage items and propose that passengers with few pieces should be seated close to the entry. Zeineddine (2017) emphasizes the importance of groups when traveling by aircraft and proposes a method whereby all group members should board together, assuming a minimum of individual interferences in the group. Bachmat et al. (2009) demonstrate with an analytical approach that boarding efficiency is linked to the aircraft interior design (seat pitch and passengers per row), so Chung (2012) and Schultz et al. (2013b) address the aircraft seating layout and indicate that alternative designs could significantly reduce the boarding time for both single- and twin-aisle configuration. Fuchte (2014) addresses aircraft design and, in particular, the impact of aircraft cabin modifications with regard to the boarding efficiency. Schmidt et al. (2015, 2017) evaluate novel aircraft layout configurations and seating concepts for single- and twin-aisle aircraft with 180–300 seats. The innovative approach to dynamically changing the cabin infrastructure through a Side-Slip Seat is evaluated by Schultz (2017c, 2017d). In the context of calibration (input parameter) and validation (simulation results), few experimental tests are conducted. Steffen and Hotchkiss (2012) tested airplane boarding methods in a mock Boeing 757 fuselage. Kierzkowski and Kisiel (2017) provide an analysis covering the time needed to place items in the overhead bins depending on the availability of seats and occupancy of the aircraft. Gwynne et al. (2018) perform a series of small-scale laboratory tests to help quantify individual passenger boarding and deplaning movement considering seat pitch, hand luggage items, and instructions for passengers. Schultz (2017a, 2018a) provides a set of operational data including classification of boarding times, passenger arrival times, time to store hand luggage, and passenger interactions in the aircraft cabin as a fundamental basis for boarding model calibration. Miura and Nishinari (2017) conducted an experiment to understand how passengers assessed boarding/deboarding times.

If the research is aimed at finding an optimal solution for the boarding sequence, evolutionary/genetic algorithms are used to solve the complex problem (e.g. Li et al. 2007; Wang and Ma 2009; Soolaki et al. 2012; Schultz 2017d). In this context, neural network models have gained increasing popularity in many fields and modes of transportation research due to their parameter-free and data-driven nature. Reitmann and Nachtigall (2017) focus on a performance analysis of the complex air traffic management system. Maa et al. (2015) apply a Long Short-Term Neural Network to capture nonlinear traffic dynamic and to overcome issues of back-propagated error. Deep learning techniques are investigated by Zhang et al. (2018) to detect traffic accident from social media data with Deep Belief Network and Long Short-Term Memory approaches. These techniques are also used to predict traffic flows, addressing the sharp nonlinearities caused by transitions between free flow, breakdown, recovery and congestion (Lv et al. 2015; Polson and Sokolov 2017). Zhou et al. (2017) proposes a Recurrent Neural Network based microscopic car following model that is able to accurately capture and predict traffic oscillation, where Zhong et al. (2017) aims at an online prediction model of non-nominal traffic conditions.

1.2 Scope and Structure of the Document

This paper provides an approach to predict the aircraft boarding progress, which will add a significant benefit to aircraft turnaround operations (e.g. precise scheduling due to reduction of uncertainties). We use a calibrated stochastic boarding model, which covers individual passenger behaviors and operational constraints (Schultz et al. 2008, 2013), to provide reliable data about the boarding progress (Schultz 2018b). The data is input for an Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber 1997) model, which is trained with time series (boarding simulation data). The stochastic boarding model and corresponding complexity metric (time series) are briefly introduced in section 2. In section 3, general principles of the machine learning approach are discussed, followed by a test set up to demonstrate prediction capabilities of our machine learning approach. The paper closes with a summary and outlook.

2 AIRCRAFT BOARDING MODEL

For the application of the machine learning approaches, a reliable dataset is needed, which provides a time-based, comprehensive description of current aircraft boarding progress. Today, there are no appropriate information are available about the boarding progress. Therefore, a stochastic boarding model is used, which covers the dynamic passenger behavior (Schultz 2014;Schultz 2018c) and is calibrated and validated with data from field measurements (deviations <5%, see Schultz 2017a, 2018a). This model provides detailed information about passenger interactions during the boarding progress. To enable a prediction of the boarding time, these information are aggregated in a complexity metric (discrete time series of boarding progress). Since the stochastic boarding model and the complexity metric are already described in Schultz (2018b, 2018c, 2018d), only a brief introduction is provided to underline the common principles of modelling passenger movements and evaluation of boarding progress.

In the context of aircraft boarding, passenger movement is assumed to be a stochastic, forward-directed, one-dimensional and discrete (time and space) process. Therefore, the aircraft cabin layout is transferred into a regular grid with aircraft entries, aisle(s) and passenger seats (Airbus A320 as reference, Airbus 2017). The regular grid consists of equal cells with an edge length of 0.4 m, whereas a cell can either be free or occupied by exactly one passenger. The boarding process consists of a simple set of rules for passenger tasks: a) enter at assigned aircraft door, b) move forward along the aisle from cell to cell until reaching the assigned seat row, and c) store hand luggage and take the seat. The passenger movement only depends on the state of the next cell (free or occupied). The hand luggage storage and seat taking are stochastic processes and depend on individual number of hand luggage items and seat constellations in seat row, respectively. An operational scenario is defined by underlying seat layout, number of passengers to board, arrival rate of passengers, boarding strategy, and compliance of passengers in following boarding strategy. Finally, each boarding scenario is simulated 100,000 times to achieve reliable calculation results for the average boarding time.

2.1 Complexity Metric

Passenger interactions during the aircraft boarding are transferred into a quantitative (complexity) metric to evaluate the boarding progress (Schultz 2018b) based on the already realized boarding progress, position of used seats, and actual passenger sequence. The interactions in the seat row are quantified by seat positions (window, middle, aisle) and corresponding operational seat conditions C_{seat}: not used/ not booked ($C_{seat} = 0$), free ($C_{seat} = 1$), and occupied ($C_{seat} = 2$). Using an A320 as reference (three seats per side), a trinary aggregation of individual seat conditions per row (left or right) results in 27 distinct values for $C_{row,\ left|right}$ (1), where c represents the number of different seat conditions ($c = 3$). As an example, if all seats in a row are occupied C_{row} is 26 (= 3^0 x **2** + 3^1 x **2** + 3^2 x **2**). Each C_{row} condition results in a specific number of interactions between passengers and depends on the seat position of the arriving passenger.

$$C_{row,\ left|right} = c^0\ C_{aisle} + c^1\ C_{middle} + c^2\ C_{window} \tag{1}$$

If, however, the aisle seat is occupied, the next arriving passenger demands a seat shuffle with a minimum of four movements (Bazargan 2007; Schultz et al. 2008): the passenger in the aisle has to step out, the arriving passenger steps into the row (two steps) and the first passenger steps back into the row. Thus, the interference potential P_r of a specific C_{row} condition is defined as the expected value of passenger movements (interactions) derived from all probable future seat row conditions (equally distributed). Figure 1 exhibits different seat row conditions, stepwise transition and corresponding development of the interference potential P_r (green circles). Here, P_r develops over the time and decreases continuously if the seat row is filled up window seat first, followed by middle and aisle seat ($P_{r,\ step\ 0} = 8.3$, $P_{r,\ step\ 1} = 3.5$, $P_{r,\ step\ 2} = 1$). This development results in only three passenger movements, since each passenger can directly step into the row without any interactions.

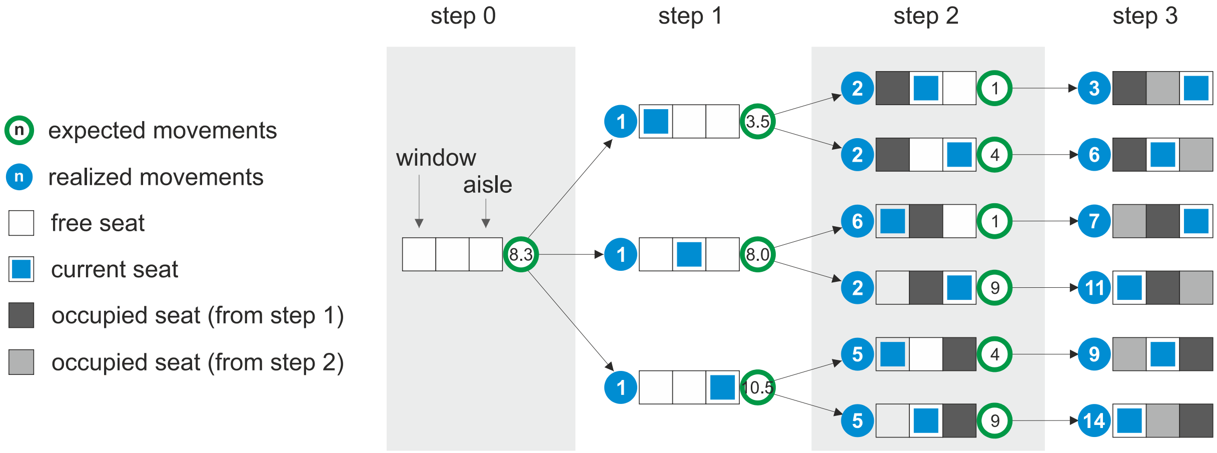

Figure 1: Seat row progress (aisle seat right, window seat left) indicating movements/interactions expected (green circle) and realized (blue bubble).

If other passengers are hindered in reaching their corresponding seat row and queue in the aisle, the negative effect of high P_r values increases and results in additional waiting times and longer boarding time. This effect depends on the number of passengers who have to pass this row or wait in the queue to reach the seats located in front of this row and is additionally covered in the aggregated aircraft–wide interference potential P (see Schultz 2017b, 2018b). Finally, these indicators address the major, time-dependent drivers of boarding: progress of seat load $1\text{-}f_{sl}$ (passengers seated) and interference potential P (blocked seat rows). Figure 2 exhibits this complexity metric using a fast and slow boarding scenario.

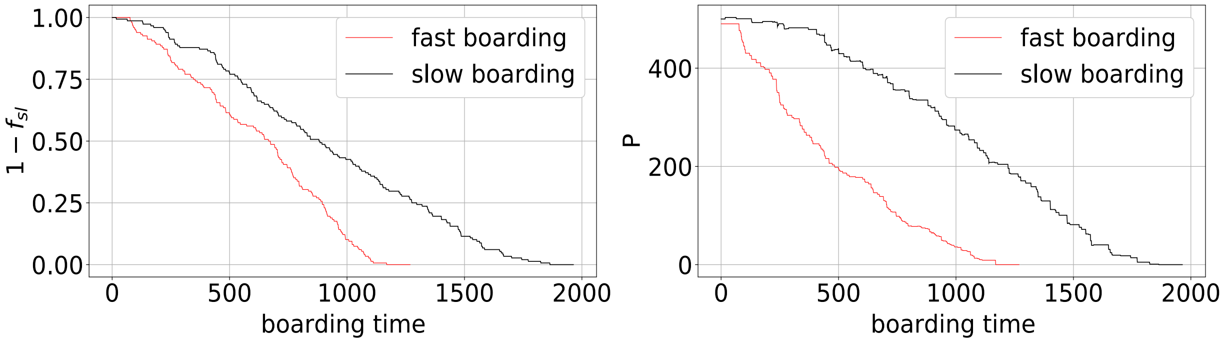

Figure 2: Complexity metric: seat load (left) and interference potential (right).

3 MACHINE LEARNING

Time predictions can be applied using advanced statistical procedures, such as neural networks, to overcome resulting complexities and dynamic effects during passenger boarding. This section provides an overview about data transformation techniques to convert time series from boarding events (simulated or real) to appropriate input datasets for machine learning (Yu et al. 2006), followed by a brief description of neural network structures and introduction of Long Short-Term Memory (LSTM) model.

3.1 Data Transformation

Neural networks use supervised learning methods, like the majority of machine learning approaches, and compare the predicted output against a reference value. For a given set of input variables $\{x(t)\}$ and output variables $\{y(t)\}$ an algorithm learns a functional dependency between input and output. The goal is to efficiently approximate the real (unknown) functional correlation and to predict output variables $y \in \{y(t)\}$ by just having input variables $x \in \{x(t)\}$. In the context of passenger boarding, a sequence of observations is given as a dataset of time series demonstrating the progress of boarding. Traditionally, time series are differentiated into univariate (observation of one single variable) and multivariate (observation of datasets) series. In the current case of aircraft boarding, the multivariate time series consists of interference potential P and seat load progress $1\text{-}f_{sl}$ provided by a validated simulation model (operational data from actual boarding events will be available in the future, if a connected aircraft cabin is realized). The application of the boarding model provides discrete time values, which could be used as multi-layer input considering current and past values. Since classical statistical methods do not perform well for this given problem, the application of complex and intelligent machine learning approaches is favored. As an example, the boarding dataset could be transformed to a supervised learning problem using P and $1\text{-}f_{sl}$ with a window size of one, as a multi-to-one time series (three input features (X) and one output value (Y), see Table 1, cf. Graves et al. (2007)). In order to follow up a multi-step prediction, the prediction datasets of the network are adjusted online (during active phase of application). Thus, we ensure that long-term predictions are made over different steps of time and train the network so that it is able to handle time-carried calculation errors.

Table 1: Multi-to-one time series using P and seat load progress ($1\text{-}f_{sl}$).

Time	**Input A**	**Input B**	**Input C**	**Output**
t	X_A	X_B	X_C	Y
0	-	-	$P(0)$	$1 - f_{sl}\ (1)$
1	$P(0)$	$1 - f_{sl}$ (1)	$P(1)$	$1 - f_{sl}\ (2)$
2	$P(1)$	$1 - f_{sl}$ (2)	$P(2)$	$1 - f_{sl}\ (3)$
3	$P(2)$	$1 - f_{sl}$ (3)	$P(3)$	prediction

3.2 Neural Network Models

A standard recurrent neural network (RNN) computes the hidden vector sequence $h = (h(1), ..., h(T))$ and output vector sequence $y = (y(1), ..., y(T))$ for a given input sequence $x = (x(1), ..., x(T))$ as shown in (2) and (3). The looped structure of RNN allows information to be passed from one step to the next. The term W denotes a weight matrix to the corresponding connections (e.g. W_{hy} is the hidden-output weight matrix), which basically comprise the parametrization sensibilities of the network. The b terms denote bias vectors as an extraction of the threshold function (e.g. b_y as the output hidden vector) and H is the hidden layer function. Usually H is an element-wise application of the logistic sigmoid function.

$$h(t) = H\,(\,W_{hy}\,x(t) + W_{hh}\ h(t-1) + b_h\,) \tag{2}$$
$$y(t) = W_{hy}\ h(t) + b_y \tag{3}$$

Unlike classical feedforward neural networks, RNN models perform well in detecting and processing long-term correlations of inputs and outputs. Figure 3 depicts the effect of input vectors $x(0)$ and $x(1)$ to the hidden output $h(t{+}1)$. Such interdependencies, delayed over certain time steps, are essential part of a multivariate data analysis, as the dynamic of the system needs to be described through the influence of indicators to each other. Long Short-Term Memory (LSTM) networks address the problem of vanishing gradients of RNN by splitting in four inner gates and building so-called memory cells to store information in a long-range context. LSTM networks are explicitly designed to avoid the long-term dependency problem. Instead of having a single neural network layer, the inner workings of LSTM modules are divided into four gates. The LSTM structure is implemented through the following equations (4-7).

$$h\ i(t) = \sigma\ (\ Wxi\ x(t) + Whi\ h(t\text{-}1) + Wci\ c(t\text{-}1) + bi\ ) \tag{4}$$

$$f(t) = \sigma\ (\ Wx\,f\ x(t) + Wh\,f\ h(t\text{-}1) + Wc\,f\ c(t\text{-}1) + bf\ ) \tag{5}$$

$$c(t) = f\ (t) \ast c(t\text{-}1) + i(t) \ast tanh\ (Wxc\ x(t) + Whc\ h(t\text{-}1) + bc\ ) \tag{6}$$

$$o(t) = \sigma\ (\ Wxo\ x(t) + Who\ h(t\text{-}1) + Wco\ c(t) + bo\ ) \tag{7}$$

Here, σ and *tanh* represent the specific, element-wise applied activation functions of the LSTM, and *i*, *f* , *o*, *c* denote the inner-cell gates, respectively the input gate, forget gate, output gate, and cell activation vectors. Gate *c* needs to be equal to the hidden vector *h*. As LSTM models are RNN based models, the overall composition is similar with regards to transform an input set to an output set through a module *A*. In LSTM models *A* consists of a cell memory (information storage) with four gates: computing input, updating cell, forget about certain information and setting output (Gers et al. 2000). To train RNN structures, a backpropagation through time algorithm is applied (Hochreiter et al. 2001). It is used for calculation of the gradients of the error with respect to weight matrices {W}. For a given batch size, which sets the amount of data after a gradient, an update is performed and the neuronal network can adjust its parametrization to retrieve given target values. With the capability to "store" knowledge, LSTMs overcome RNN deficiency to handle vanishing gradients (a backpropagation of *tanh* values near 0 or 1 over time might lead to a blowing-up or disappearing). RNN-based models are characterized by the number of hidden layers ($n_{\text{hiddenlayer}}$), the number of samples propagated through the network for each gradient update (batch-size), the number of trained epochs (n_{epoch}) and the learning rate (η).

4 SIMULATION FRAMEWORK AND APPLICATION

The input data for the LSTM model is provided by our introduced boarding model. We implemented the given RNN structures in Python 3.6 using the open-source deep learning library Keras 2.1.3 (frontend) with open-source framework TensorFlow 1.5.0 (backend) and Scipy 1.0.0 (routines for numerical integration and optimization). Training and testing were performed on GPU (NVIDIA Geforce 980 TI) with CUDA as parallel computing platform and application programming interface.

4.1 Simulation Framework

The simulation framework consists of fundamental components: layers (LSTM, concatenate, input, dense, dropout), optimizers, and metric/loss functions. The structure of the implemented LSTM depends on the given scenario. For a one-dimensional application, one input layer is added to the model, followed up by the LSTM recurrent layer, dense layer, and dropout layer. Figure 3 depicts the network structure for both cases one-to-one (solid) and multi-to-multi regression (dotted). Each layer consists of specific tasks in the framework.

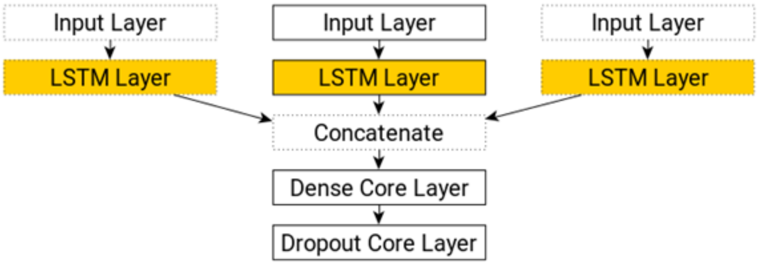

Figure 3: Layer design of simulation framework.

The LSTM (recurrent) layer represents the specific machine learning model. The input layer is a Keras tensor object from the underlying backend (TensorFlow) and contains certain attributes to build a Keras model (knowing only in-/outputs). The concatenate layer merges several inputs (list of tensors, with same shape) and returns a single tensor. The dense layer is a regular densely-connected neural network layer and implements the fundamental output operation: output = *act* (dot(input, *kernel*) + bias), where *act* is the element-wise activation function, *kernel* is a weights matrix created by the layer, and *bias* is a vector created by the layer. The dropout layer provides a regularization technique, which prevents complex co-adaptations on training data in neural networks (overfitting), by dropping a number of samples in the network (dropout rate). An optimizer is one of the two arguments required to finally compile the Keras model. Two optimizers are available: AdaGrad (adaptive gradient algorithm), modified stochastic gradient descent approach with per-parameter learning rate (Duchi et al. 2011), and Adam (Adaptive Moment Estimation), family of sub-gradient methods that dynamically incorporate knowledge of the geometry of the data observed in earlier iterations (Kingma and Ba 2014) . The optimizers are parametrized with learning rate $\eta = 0.01$ and $\eta = 0.001$ for AdaGrad and Adam respectively. The loss function (optimization score function) of our model is the mean squared error (MSE).

4.2 Scenario Definition

In Table 2 an overview is given about the analyzed scenarios, focusing on the prediction of boarding progress of *random* and *individual* (Steffen 2008) boarding strategies. In scenario A, the prediction of *random* boarding is based on learning of *random* boarding scenarios. In scenario B, the progress of *individual* boarding is predicted on the basis of all available boarding data sets (cf. Schultz 2018c). As input values P, $1\text{-}f_{sl}$, or both are used to predict the output $1\text{-} f_{sl}$. Both scenarios are computed for a given set of start times: 300s, 400s, and 600s after boarding starts. Supervised learning demands for a separation of datasets into training, test and (obligatory) validation data. In our scenario analysis, we have 25,000 separate boarding results for *random* and *individual* boarding, and 100,000 boarding results from other strategies. We use 1,000 randomly chosen boarding events to train LSTM and 100 randomly chosen non-compiled /non-trained boarding events to evaluate the prediction.

Table 2: Scenario definition.

scenario	boarding strategy learned	boarding strategy predicted	input from complexity metric	prediction start time t (s)
A	*random*	*random*	$\{P$, $1\text{-}f_{sl}$, $[P$, $1\text{-} f_{sl}]\}$	{300, 400, 500}
B	all (*random*, *block*, *back-to-front*, *outside-in*, *reverse pyramid*, *individual*)	*individual*		

4.3 Results

We trained the model within 18 different variants: two main scenarios, each with three different input sets split into three different starting points. Similar experiments demonstrate several clues concerning useful combinations of certain optimizers and network structures depending on the amount of input data (Reitmann and Nachtigall 2017). The number of hidden neurons depends on the given task and the prepared data. It needs to be balanced to avoid both underfitting (not enough complexity of the model to rebuild the data) and overfitting (to similar to training data). To adjust this hyper parameter for a network, we have chosen (8) the recommendation provided by Hagan et al. (2014) as follows:

$$N_h{=}N_s \,/\, (\alpha * (N_i + N_o)), \tag{8}$$

with N_h = number of hidden neurons, N_i = number of input neurons, N_o = number of output neurons, N_s = number of samples in training data set, and α = an arbitrary scaling factor usually 2-10. For the multi-dimensional case N_i is 2, N_o is 1 and N_s is 1800 (exemplary number of boarding dataset length). The size of α depends on whether one needs a more general model or prevents overfitting. As we need to balance both, we chose α = 5 for the multidimensional experiments, which results in N_h = 120 for the multi-dimensional LSTM. Due to overfitting problems within the univariate approach with N_i = 1, we decreased the actual N_h for α = 10 (N_h = 90) to 40. AdaGrad works quite well with a more complex network structure, as its learning rate η is 10 times lower than η of Adam, which improves gradient-based learning in multi-branched inlays. For both cases, uni- and multi-variate, we trained the model 20 times, which results 20,000 training cycles. The window size is set to 25 and 5 for uni-variate and multi-variate respectively. The batch size (number of samples per gradient update) was set to 50 to balance computation time and accuracy. The model worked with a dropout rate of 0.5% for one-dimensional, 2.0% for multi-dimensional applications (see Figure 4).

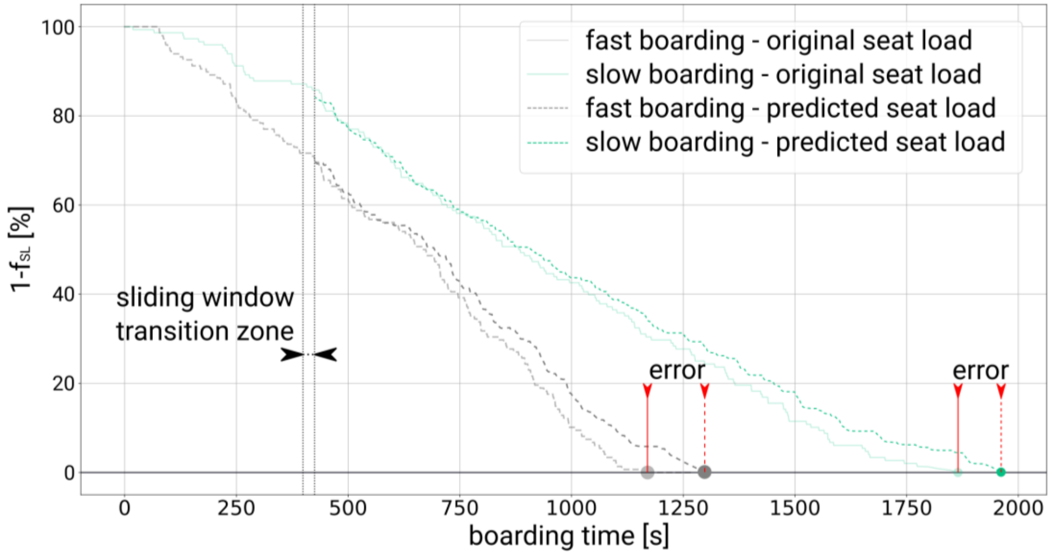

Figure 4: Prediction of seat load progress with LSTM model for two exemplary processes (start at 400 s).

Table 3 provides an overview of the accuracy of each single model to the corresponding scenario including the computation time. The accuracy is measured as mean error value of 100 tested scenarios after training phase (1.000 sample sets). In table cells are color coded, where red stands for MSE (learning error) > 65 %, green if MSE < 25%, and yellow is in between these values. As common statistical regression models would gradually suffer from calculation errors, neural network models receive the knowledge to end at point 0 (end of boarding) in every case of training.

Table 3: Results of computations scenarios A and B.

scenario	**input (uni-variate)**	**input (uni-variate)**	**input (multi-variate)**
A	***[1 – f_{sl}]***	***[P]***	***[1 – f_{sl}, P]***
300	NaN t = 1h 28min	NaN t = 1h 43min	289.8s t =3h 02min
400	536.1s t = 0h 30min	NaN t = 1h 01min	143.7s t = 2h 38min
500	301.8s t = 0h 32min	NaN t = 0h 49min	122.3s t = 2h 12min
B	***[1 – f_{sl}]***	***[P]***	***[1 – f_{sl}, P]***
300	603.1s t = 1h 13min	NaN t = 1h 19min	168,1s t = 7h 38min
400	292.1s t = 1h 12min	389.9s t = 1h 10min	77.3s t = 8h 01min
500	246.0s t = 1h 08min	412.4s t = 1h 11min	73.3s t = 7h 54min

An insufficiently trained model might show bifurcations and chaotic behavior during the boarding process (red colored cells or *NaN* results), but would tend to reach point 0 at the end in any case. In our case, uni-variate inputs are not sufficient, only multi-variate input results in reliable output values. Figure 5 shows time differences at specific progress steps, using 400 s as start time for prediction. The box-plots of time differences refer to values illustrated in Table 3, using an interquartile ratio of 0.7.

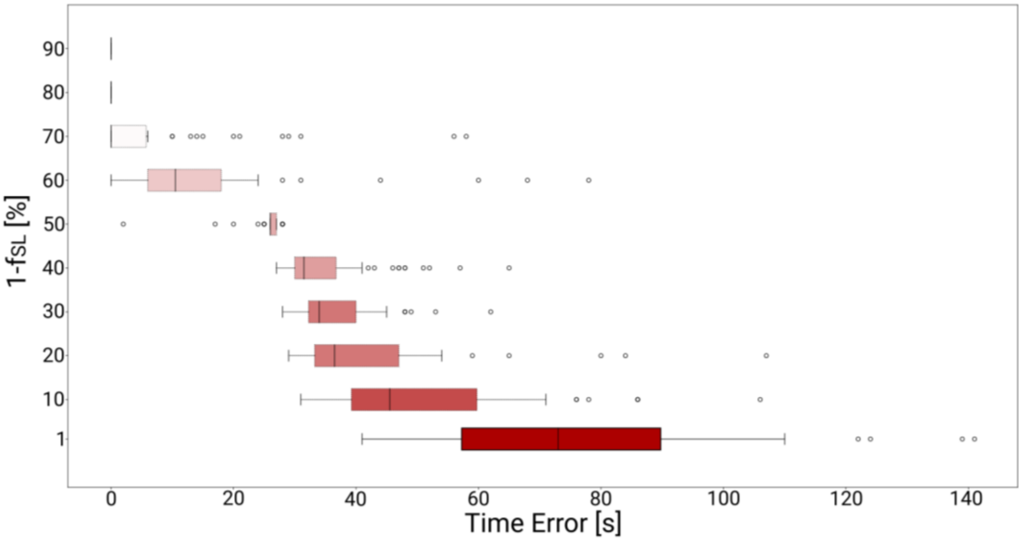

Figure 5: Scenario B (all data sets as input – *individual* boarding to predict) – time difference.

The red colored box include the last computed steps of the simulations (99% progress). As expected, the time variation in random boarding is significantly higher than in individual boarding. Both predictions result in a positive and nearly symmetric offset over time. The major difference between scenario A and B is that in scenario A the LSTM is trained with corresponding data sets. In the scenario B uses all available data sets for training (only excluding data sets for testing). In doing so, the computational time for training increases significantly. Due to the higher bandwidth of operational progresses in scenario B the LSTM model is able to identify more interdependencies during the training process, which results in a more precise prediction.

5 SUMMARY AND OUTLOOK

Future 4D aircraft trajectories demand the comprehensive consideration of environmental, economic, and operational constraints. A reliable prediction of all aircraft-related processes along the specific trajectories is essential for punctual operations. To provide a reliable prediction of the turnaround, the critical path of processes has to be managed in a sustainable manner. This paper provides an approach to appropriately predict the aircraft boarding progress, which will add a significant benefit to aircraft turnaround operations (e.g. precise scheduling due to reduction of uncertainties). We use a calibrated stochastic boarding model, which provides current status information about the boarding progress. This status contains mainly seat load (percentage of seated passengers) and interference potential (complexity measure considering already used seats). These data are input for an Long Short-Term Memory (LSTM) model, which is trained with boarding simulation data (discrete time series) and enables an prediction of the final boarding time. We demonstrate that the proposed complexity metric (multi-variate input) is a necessary element to predict the aircraft boarding progress. A closer look to differences between boarding progress and prediction shows an inherent positive offset. If the offset is assumed as constant, predictions show deviations smaller than 20 s. Since operational requirements demands for qualitative information about the probability, if a given target time will be reached (boarding ends before) or missed, we see our LSTM prediction model as a promising candidate for extended investigations (Schultz and Reitmann 2018).

REFERENCES

Airbus. 2017. Airbus A320 Aircraft Characteristics - Airport and Maintenance Planning.

Bachmat, E., D. Berend, L. Sapir, S. Skiena, and N. Stolyarov. 2009. "Analysis of Aircraft Boarding Times". *Operations Research* 57:499–513.

Bazargan, M. 2007. "A Linear Programming Approach for Aircraft Boarding Strategy". *Europ. J. of Operational Research* 183(1):394–411.

Bronsvoort, J., G. McDonald, R. Porteous, and E. Gutt. 2009. "Study of Aircraft Derived Temporal Prediction Accuracy using FANS". *ATRS Conference*, Abu Dhabi.

Chung, Ch. A. 2012. "Simulation Design Approach for the Selection of Alternative Commercial Passenger Aircraft Seating Configurations". *J. of Aviation Technology and Engineering* 2:100–104.

Cook, A.J. and G. Tanner. 2015. "European airline delay cost reference values". Technical Report. EUROCONTROL Performance Review Unit.

Duchi, J., E. Hazan, Y. Singer. 2011. "Adaptive subgradient methods for online learning and stochastic optimization". *JMLR* 12: 2121–2159.

Eurocontrol. 2015. "Performance Review Report - An Assessment of Air Traffic Management in Europe During the Calendar Year 2015".

Fricke, H. and M. Schultz. 2008. "Improving Aircraft Turn Around Reliability". In *Proceedings of 3rd ICRAT*, 335–343., Fairfax

Fricke, H. and M. Schultz. 2009. "Delay Impacts Onto Turnaround Performance". In *Proceedings of 8th USA/Europe ATM Seminar*, Napa.

Fuchte, J. 2014. *Enhancement of Aircraft Cabin Design Guidelines with Special Consideration of Aircraft Turnaround and Short Range Operations*. Ph.D. Thesis, TU Hamburg-Harburg.

Gers, F.A., J. Schmidhuber, and F. Cummins. 2000. "Learning to Forget: Continual Prediction with LSTM". *Neural Computation* 12:2451–2471.

Graves, A., S. Fernandez, and J. Schmidhuber. 2007. "Multi-Dimensional Recurrent Neural Networks". In *Proc. International Conf. on Artificial Neural Networks (ICANN-2007)* 4668:865–873, Porto.

Gwynne, S.M.V., U. Senarath Yapa, L. Codrington, J.R. Thomas, S. Jennings, A.J.L. Thompson, and A. Grewal. 2018. "Small-scale trials on passenger microbehaviours during aircraft boarding and deplaning procedures". *J. Air Tr. Manag.* 67:115–133.

Hagan, M.T., H.B Demuth, M.H. Beale, O. De Jesús. 2014. *Neural Network Design*. 2nd Edition.

Hochreiter, S. and J. Schmidhuber. 1997. "Long Short-Term Memory". *Neural Computation* 9(8):1735–1780.

Hochreiter, S., Y Bengio, P Frasconi, J Schmidhuber. 2001. *Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies*. Wiley-IEEE Press.

Jaehn, F., and S. Neumann. 2015. "Airplane Boarding". *European Journal of Operational Research* 244(2):339–359.

Kierzkowski, A. and T. Kisiel. 2017. "The Human Factor in the Passenger Boarding Process at the Airport". *Procedia Engineering* 187:348–355.

Kingma, D.P. and J. Ba. 2014. "Adam: A method for stochastic optimization". In *Proceedings of the 3rd ICLR*, Banff.

Li, Q., A. Mehta, and A. Wise. 2007. "Novel approaches to airplane boarding". *UMAP Journal* 28(3): 353–370.

Lv, Y., Y. Duan, W. Kang, Z. Li, and F.Y. Wang. 2015. "Traffic flow prediction with big data: a deep learning approach". *IEEE Trans. Intell. Transp. Syst.* 16:865-873.

Maa, X., Z. Tao, Y. Wang, H. Yu, and Y. Wang. 2015. „Long short-term memory neural network for traffic speed prediction using remote microwave sensor data", *J. of Trans. Res. Part C: Emerg. Techn.* 54:187–197.

Milne, R.J., and A.R. Kelly. 2014. "A New Method for Boarding Passengers onto an Airplane". *J. Air Transp. Manag.* 34:93–100.

Milne, R.J., and M. Salari. 2016. "Optimization of Assigning Passengers to Seats on Airplanes Based on Their Carry-on Luggage". *J. Air Transp. Manag.* 54:104–110.

Mirza, M. 2008. "Economic impact of airplane turn-times". *AERO Quart.* 4:14–19.

Miura, A. and K. Nishinari. 2017. "A passenger distribution analysis model for the perceived time of airplane boarding/deboarding, utilizing an ex-Gaussian distribution". *J. Air Transp. M.* 59:44–49.

Mueller, E.R. and G. B. Chatterji. 2002. "Analysis of Aircraft Arrival and Departure Delay". *AIAA Conference*, Los Angeles.

Niklaß, M., B. Lührs, V.Grewe, K.Dahlmann, T.Luchkova, F.Linke, and V.Gollnick. 2017. "Potential to reduce the climate impact of aviation by climate restricted airspaces". *Transport Policy*.

Nyquist, D.C., and K.L. McFadden. 2008. "A study of the airline boarding problem". *J. Air Transp. Manage.* 14:197–204.

Polson, N.G. and V.O. Sokolov. 2017. "Deep learning for short-term traffic flow prediction". *J. of Trans. Res. Part C: Emerg. Techn.* 79:1–17.

Qiang, S.J., B. Jia, D.-F. Xie, and Z.-Y. Gao. 2014. "Reducing Airplane Boarding Time by Accounting for Passengers' Individual Properties: A Simulation Based on Cellular Automaton". *Journal of Air Transp. Manage.* 40:42–47.

Reitmann, S., and K. Nachtigall. 2017. "Applying Bidirectional Long Short-Term Memories to Performance Data in Air Traffic Management for System Identification". *Lect. Notes in Comp. S.* 10614, 528-536.

Rosenow, J., M. Lindner, and H. Fricke. 2017a. "Impact of climate costs on airline network and trajectory optimization: a parametric study". *CEAS Aeronautical Journal* 8(2):371–384

Rosenow, J., H. Fricke, and M. Schultz. 2017b. "Air traffic simulation with 4d multi-criteria optimized trajectories". In *Proceedings of WSC 2017*, 2589-2600, Las Vegas.

Rosenow, J., H. Fricke, T. Luchkova, and M. Schultz. 2018. "Minimizing contrail formation by rerouting around dynamic ice-supersaturated regions". *AAOAJ* 2(3): 105–111.

Schmidt, M. 2017. "A Review of Aircraft Turnaround Operations and Simulations". *Progress in Aerospace Sciences* 92:25–38.

Schmidt, M., P. Heinemann, and M. Hornung. 2017. "Boarding and Turnaround Process Assessment of Single- and Twin-Aisle Aircraft". *AIAA* 2017-1856, Grapevine.

Schmidt, M., P. Nguyen, and M. Hornung. 2015. "Novel Aircraft Ground Operation Concepts Based on Clustering of Interfaces". *SAE Technical Paper* 2015-01-2401, Seattle.

Schultz, M. 2014. "Stochastic Transition Model for Pedestrian Dynamics". In *Pedestrian and Evacuation Dynamics 2012*, 971–986. Springer.

Schultz, M. 2017a. *Aircraft Boarding - Data, Validation, Analysis*. In *Proceedings of 12th USA/Europe ATM Seminar*, Seattle.

Schultz, M. 2017b. "The seat interference potential as an indicator for the aircraft boarding progress". *SAE Technical Paper* 2017-01-2113, Fort Worth.

Schultz, M. 2017c. "Faster aircraft boarding enabled by infrastructural changes". In *Proceedings of WSC 2017*, 2530–2541, Las Vegas.

Schultz, M. 2017d. "Dynamic Change of Aircraft Seat Condition for Fast Boarding". *J. of Trans. Res. Part C: Emerg. Techn.* 85:131–147.

Schultz, M. 2018a. "Field Trial Measurements to Validate a Stochastic Aircraft Boarding Model". *Aerospace* 5(1):27

Schultz, M. 2018b. "A metric for the real-time evaluation of the aircraft boarding progress". *J. of Transp. Res. Part C: Emerg. Techn.* 86:467-487.

Schultz, M. 2018c. “Implementation and Application of a Stochastic Aircraft Boarding Model”. *J. of Transp. Res. Part C: Emerg. Techn.* 90:334-349

Schultz, M. 2018d. “Fast Aircraft Turnaround Enabled by Reliable Passenger Boarding”. *Aerospace* 5(1):8.

Schultz, M., C. Schulz, and H. Fricke. 2008. “Efficiency of Aircraft Boarding Procedures”. In *Proceedings of 3rd ICRAT*, 371–377, Fairfax.

Schultz, M., T. Kunze, and H. Fricke. 2013. “Boarding on the Critical Path of the Turnaround”. In *Proceedings of 10th USA/Europe ATM Seminar*, Chicago.

Schultz, M. and S. Reitmann. 2018. “Machine learning approach to predict aircraft boarding”. *J. of Transp. Res. Part C: Emerg. Techn.* (review)

Soolaki, M., I. Mahdavi, N. Mahdavi-Amiri, R. Hassanzadeh, and A. Aghajani. 2012. “A new linear programming approach and genetic algorithm for solving airline boarding problem”. *Appl. Math. Mdl.* 36:4060–4072.

Steffen, J.H. 2008. “Optimal Boarding Method for Airline Passengers”. *Journal of Air Transportation Management* 14:146–150.

Steffen, J.H., and J. Hotchkiss. 2012. “Experimental test of airplane boarding methods”. *J. Air Trans. Manage.* 18:64–67.

Tielrooij, M., C. Borst, M.M. van Paassen, and M. Mulder. 2015. “Predicting Arrival Time Uncertainty from Actual Flight Information”. In *Proceedings of 11th USA/Europe ATM Seminar*, Lisbon.

Wang, K. and L. Ma. 2009. “Reducing boarding time: Synthesis of improved genetic algorithms”. In *Proceedings of the 2009 WEBIST*, 359–362, Lisbon.

Yu, L., S. Wang, and K. Keung Lai. 2006. “An integrated data preparation scheme for neural network data analysis”. *IEEE Trans. Knowl. Data Eng.* 18(2):217–230.

Zeineddine, H. 2017. “A Dynamically Optimized Aircraft Boarding Strategy”. *Journal of Air Transportation Management* 58:144–151.

Zhang, Z., Q. He, J. Gao, and M. Ni. 2018. “A deep learning approach for detecting traffic accidents from social media data”. *J. of Transp. Res. Part C: Emerg. Techn.* 86, 580–596.

Zhong, R.X., J.C. Luo, H.X. Cai, A. Sumalee, F.F. Yuan, and A.H.F. Chow. 2017. “Forecasting journey time distribution with consideration to abnormal traffic conditions”. *J. of Transp. Res. Part C: Emerg. Techn.* 85:292–311.

Zhou, M., X. Qu, and X. Li. 2017. “A recurrent neural network based microscopic car following model to predict traffic oscillation”, *J. of Transp. Res. Part C: Emerg. Techn.* 84:245–264.

AUTHOR BIOGRAPHY

MICHAEL SCHULTZ is Head of the Air Transportation Department of the Institute of Flight Guidance at the German Aerospace Center (DLR), Brunswick, Germany. He holds a diploma degree in business and engineering and a Ph.D. in aviation technologies and logistics from the Technische Universität Dresden, Germany. His research interests include stochastic, agent-based simulations and their applications in the transportation sector, particularly in air traffic and airport management. His e-mail address is: michael.schultz@dlr.de.

STEFAN REITMANN is Ph.D. student at the German Aerospace Center (DLR), Brunswick, Germany. In 2015 he received his diploma in Traffic Engineering at the Dresden University of Technology (TU Dresden) with focus on traffic flow sciences. In 2016 and 2017 he was part of the scientific exchange program of the German-Russian Interdisciplinary Science Center (G-RISC) and stayed at the State University St. Petersburg (SPbU). Associated with this research he received the Young Russian-German Scientist Award in 2016. His research focus is on machine learning and big data analysis, especially for the usage of neural networks in traffic sciences. In addition to his Ph.D. thesis he has been studying Informatics at the University of Hagen since 2015. His e-mail address is stefan.reitmann@dlr.de.