Transportation Research Part C 98 (2019) 391–408

Contents lists available at ScienceDirect

Transportation Research Part C

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

Machine learning approach to predict aircraft boarding☆

Michael Schultz*, Stefan Reitmann

Institute of Flight Guidance, German Aerospace Center, Braunschweig, Germany

ARTICLE INFO

Keywords:
Aircraft boarding
Air transportation
Recurrent neural network
LSTM
Progress Prediction

ABSTRACT

Reliable and predictable ground operations are essential for punctual air traffic movements. 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 used a recurrent neural network approach to predict the progress of a running boarding event. In particular, we implemented and trained the Long Short-Term Memory model. Since no operational data of the specific passenger behavior is available, we used a reliable, validated boarding simulation environment to provide data about the aircraft boarding events. First predictions show that uni-variate input (seat load progress) produces insufficient results, so we consider expected passenger interactions in the aircraft cabin as well. These interactions are aggregated to a prior-developed complexity metric and allow an efficient evaluation of the current boarding progress. With this multi-variate input, our Long Short-Term Memory model achieves appropriate prediction results for the boarding progress.

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 min 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, 2017). 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 air traffic management primarily addresses the economic and ecological impact of flight trajectories (Rosenow et al., 2017a,b, 2018; Niklaß et al., 2017) but has to include efficient aircraft ground operations as well in order to ensure an efficient aircraft trajectory over the day of operations (Rosenow and Schultz, 2018).

The aircraft turnaround on the ground consists of seven major ground handling operations at the stand: deboarding, catering, cleaning, fueling and boarding as well as the parallel processes of unloading and loading (cf. Airbus, 2017). All these handling

☆ This article belongs to the Virtual Special Issue on "AI in ATM".

* Corresponding author.
E-mail addresses: michael.schultz@dlr.de (M. Schultz), stefan.reitmann@dlr.de (S. Reitmann).

https://doi.org/10.1016/j.trc.2018.09.007
Received 31 January 2018; Received in revised form 7 September 2018; Accepted 9 September 2018
Available online 24 December 2018
0968-090X/

processes follow clearly defined procedures and are mainly controlled by ground handling, airport or airline staff (Fricke and Schultz, 2008, 2009). 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, specific (high) numbers of hand luggage items, or priority passengers (privileged boarding). To provide reliable values for the target off-block time (SESAR, 2014), which is used as a planning time stamp for the subsequently following departure procedures, all seven major turnaround processes are assessed on prediction. With the given focus on the aircraft boarding, this stochastic and passenger-controlled process makes it difficult to reliably predict the turnaround and target off-block time, even if boarding is already in progress.

Today, there are no operational data available from inside the aircraft cabin, which could provide actual status information about the current boarding progress. Data from small-scale laboratory tests (cf. Gwynne et al., 2018) or field measurement campaigns (Schultz, 2018a) only provide input to calibrate models for simulation of aircraft boarding. Therefore, a developed and validated stochastic boarding model (Schultz, 2018b) is used to provide a comprehensive set of data from inside the aircraft cabin (Schultz, 2018c). In this paper, we develop and implement a recurrent neural network approach to predict the boarding progress using this quasi-actual cabin information. It is expected that a future connected aircraft cabin will provide a sensor environment, which could be used as operational input for a real-time prediction of the boarding progress and sustainably improve the turnaround management.

1.1. Status quo

Our research is connected to three different topics: aircraft turnaround (focus on boarding), passenger behavior (complexity metric), and machine learning (recurrent neural networks). 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 (cf. Cook and Tanner, 2015). Relevant studies include, but are not limited to, the following current examples.

The aircraft turnaround, as part of the aircraft trajectory over the day of operations, has to be part of optimization strategies for minimizing flight delays (Montlaur and Delgado, 2017) and ensuring flight connection considering operational uncertainties (cf. Oreschko et al., 2011, 2012). In this context, turnaround absorbs inbound delay (Fricke and Schultz, 2009) and could enhance slot adherence at airports (Ivanov et al., 2017) or mitigate problems of push-back scheduling (Du et al., 2014). Schultz et al. (2013a) provide a microscopic turnaround model aiming at open and closed-loop process control for higher automation levels in turnaround management. Kafle and Zou (2016) focused on an inter-aircraft propagated delay, since individual delays could result in parallel demand of turnaround resources (personnel, space, and equipment). Furthermore, delayed use of infrastructure may cause excessive demand in later time frames, and both turnaround stability and resource efficiency will provide significant benefits to airline and airport operations (Fricke and Schultz, 2009; Grunewald, 2016). Okwir et al. (2017) explore turnaround performance focusing on collaborative management involving all stakeholders. Mota et al. (2017) analyze turnaround for identification of variables that affect the future performance. When airports are dealing with heavy traffic demand, operators are often pushed to the limit of their available capacity. To efficiently handle airport operations, Mirkovic et al. (2016) provide a tool for an interactive resource allocation. When introducing new aircraft using alternative energy concepts, Schmidt et al. (2016) aim at compatibility with airline operations, existing ground handling procedures and airport infrastructure requirements.

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 (2017). In the context of calibration (input parameter) and validation (simulation results), few experimental tests have been 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 (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, 2017). 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. Neural networks are able to rebuild highly complex functions and identify data interdependencies of multidimensional nature. As a network can be trained to certain kind of problems, a real-time usage of this network can be done with low

computing times. 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. Zhu et al. (2017) propose a bi-level model to solve the timetable design problem for an urban rail line, where the upper level model determines headways between trains to minimize total passenger cost and the lower level model addresses passengers' arrival times at their origin. Deep learning techniques are investigated by Zhang et al. (2018) to detect traffic accidents 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) propose a microscopic car-following model based on a Recurrent Neural Network that is able to accurately capture and predict traffic oscillation, whilst Zhong et al. (2017) aim at an online prediction model of non-nominal traffic conditions. In the context of application of machine learning approaches to the airspace and air traffic management, Cao et al. (2018) provide an evaluation of the operation complexity of airspace sectors, Verdonk Gallego et al. (2018) analyze the impact of operational factors on the vertical flight profiles, Wang et al. (2018) address the short-term trajectory prediction problem in the terminal maneuvering area, and Gerdes et al. (2016, 2018) and Standfuss et al. (2018) develop an approach for a dynamic airspace sectorization for flight-centric operations.

1.2. Scope and structure of the document

This paper provides, for the first time, an approach to predict aircraft boarding progress, which will add a significant benefit to aircraft turnaround operations (e.g. precise scheduling due to reduction of uncertainties). Since no operational data are available from inside the aircraft cabin, we use a validated stochastic boarding model as simulation environment, which covers individual passenger behaviors and operational constraints (Schultz et al., 2008, 2013a,b), to provide reliable status information about the boarding progress (Schultz, 2018c). This status contains seat load (percentage of seated passengers), interference potential (complexity measure considering already used seats), and the quality of the boarding sequence (chronological order of passenger arrival). The status information is input for a Long Short-Term Memory (LSTM, Hochreiter and Schmidhuber (1997)) model (sub type of recurrent neural networks), which is trained with boarding simulation data (time series) and enables a prediction over time (see Fig. 1).

After the introduction, in Section 2, the stochastic boarding model and the simulation environment are briefly introduced, followed by a description of general boarding progress. To evaluate the boarding progress in more detail, a developed complexity metric is presented (Schultz, 2018c), which covers specific interdependencies between passengers during luggage storing and seating process. In particular, the derived interference potential demonstrates the high impact of these interdependencies on the final boarding time and could indicate a trend as to how the boarding progress will proceed (qualitative differentiation between fast and slow boarding progress). At this point, the complexity metric seems to be a good candidate for machine learning approaches to allow for quantified prediction of boarding time. In Section 3, general principles of the machine learning approach are discussed, including the preparation of input datasets and the introduction of neural network models, with a focus on the Long Short-Term Memory (LSTM) approach. To use the prediction model in real operations, immediate access to the boarding data during the ongoing operations is mandatory. However, considering the high computing times and the system requirements, a live training system would not be applicable for real-time usage. Therefore, the neural network needs to be adjusted to a huge bandwidth of historical data of boarding events for a reliable prediction. As the LSTM model requires the same input data in the application as in the training, the complexity metric data need to be receivable in real-time (see Fig. 2).

In a field trial at Cologne/Bonn airport with Eurowings, a prototype of a connected cabin was successfully tested. Here, the necessary sensor information could be provided and transformed to a complexity value in real-time (cf. Schultz, 2018d). In Section 4, we set up a test scenario to demonstrate the prediction capabilities of our machine learning approach. We choose the *random* boarding (passengers have pre-assigned seats and enter the aircraft unconstrained, with no specific order) as a common boarding process and the *individual* boarding (each passenger has a specific position in the waiting queue at the gate, cf. Steffen (2008)) as one of the fastest boarding processes as reference implementations. With the stochastic boarding model, 150,000 boarding events are simulated to provide input for the recurrent neural network. The major part of these simulated events is used to train our LSTM model

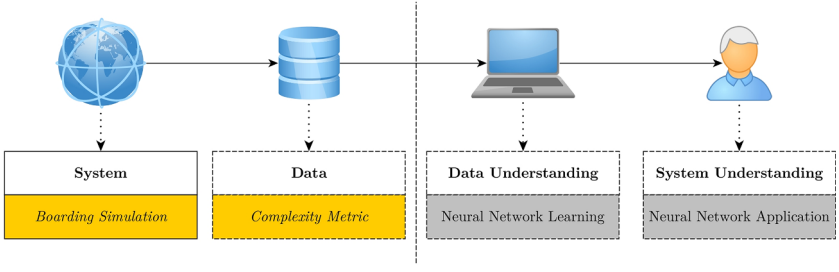

Fig. 1. General approach of data-based process analysis. System-related parts in solid lines, data-related parts in dotted lines: boarding simulation as reliable surrogate for unavailable access to real operational systems, complexity metric to extract system data, which describe system behavior, and neural network approach for both learning the complex behavior and predicting the boarding progress (cf. Reitmann and Schultz, 2018).

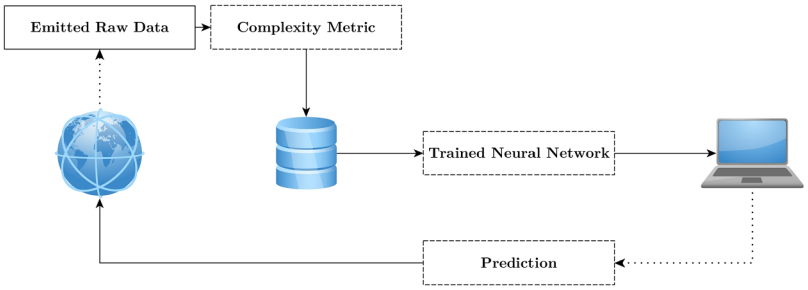

Fig. 2. Connection of the proposed model to a real-time application. The system emits raw performance data, which are aggregated in the forms the complexity metric defines. These data are input for the pre-trained model and are used to predict the boarding process ending.

and the other, learning-independent part is used for testing. We test our approach with three evaluation scenarios and demonstrate the capabilities of the LSTM model to predict the boarding progress as well as the necessity of the complexity metric as multi-variate input for the LSTM model. Finally, the paper closes with a summary and outlook.

2. Aircraft boarding model

For the application of the machine learning approaches a multi-variate dataset is needed, which provides a time-based, comprehensive description of the current aircraft boarding progress. To provide reliable status information, a stochastic boarding model is used, which covers both the dynamic behavior of passengers and operational constraints (Schultz, 2018b). The input parameters for the boarding model (passenger characteristics) are calibrated with data from field measurements and the simulation results (boarding times) are validated against real boarding events (Schultz, 2018a). Thus, this model delivers detailed information about the boarding progress, although operational airline/aircraft systems are not able to provide this information today. To provide a comprehensive status of the boarding progress, a complexity metric was developed, which covers relevant evaluation parameters: seat load progress (number of passengers seated), expected passenger interactions during boarding, and chronological order of passengers. Since the stochastic boarding model and the complexity metric are already described in Schultz (2018b,c), only a brief introduction is given to underline the common principles of the stochastic boarding model and evaluation of boarding progress. Following this brief introduction, the complexity metric is used as multi-variate input for the machine learning environment.

2.1. Stochastic passenger behavior and process model

The model provides a reliable basis for the operational progress of aircraft boarding using a calibrated stochastic movement model for the dynamic passenger behavior (Schultz, 2014; Schultz et al., 2010). This model was previously successfully adapted to reproduce dynamic passenger behavior in the airport terminal (cf. Schultz, 2010; Schultz and Fricke, 2011). 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 with 29 rows and 174 seats (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 movements: (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 mainly defined by: underlying seat layout, number of passengers to board, arrival rate of the passengers at aircraft door, number of used aircraft doors, specific boarding strategy, and conformance of passengers in following boarding strategy. In this context, the conformance rate describes several operational deviations from the intended boarding strategy caused by boarding services provided by airlines (e.g. priority boarding, 1st class seats)

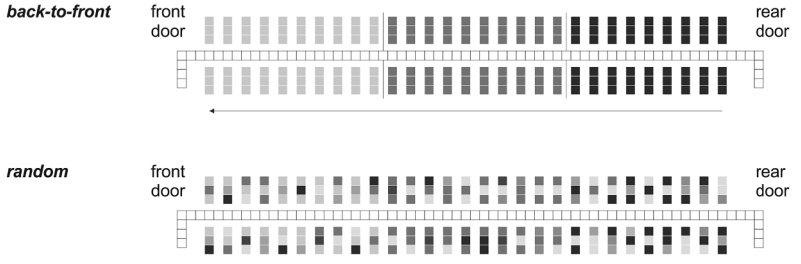

Fig. 3. Grid-based simulation environment – reference Airbus A320 layout (above) and example for *block* (here *back-to-front*) boarding strategy modelled in the simulation environment (darker seats are boarded first) (below).

or late arrival of passengers. To model different boarding strategies, the grid-based approach enables both individual assessment of seats and classification/aggregation according to the intended boarding strategy of airlines. In Fig. 3, the seats are color-coded (gray-scale) and aggregated to a superior block structure. The boarding takes place in the order of the gray-scale value (darker seats are boarded first). Boarding strategies are defined by a specific sequence of passengers (and their seats) that is mainly driven by a combination of the *block* (based on seat rows) and *outside-in* procedures (based on seat positions: window seats are boarded first, followed by middle and aisle seats). For the *block* boarding, an optimal block sequence is defined (Schultz, 2018a–d) and the *back-to-front* sequence (rear blocks boarded first) is a commonly used *block* strategy in airline operations. The *reverse pyramid* strategy is a hybrid approach of *back-to-front* and *outside-in* boarding. With the *individual* boarding, each passenger has a specific (optimal) position in the waiting queue at the gate (cf. Steffen (2008)). The *random* boarding is defined as the reference case, where passengers have pre-assigned seats and enter the aircraft unconstrained, with no specific order. For each of the six boarding strategies (*random*, *block*, *back-to-front*, *outside-in*, *reverse pyramid*, *individual*), 25,000 boarding events are simulated and available as inputs for our machine learning approach.

2.2. Simulation environment

In the simulation environment, the boarding process is implemented as follows. Depending on the seat load, a specific number of randomly chosen seats are used for boarding. For each seat, a passenger (agent) is created. The agent has individual parameters, such as number of hand luggage items, maximum walking speed in the aisle, seat position, number of hand luggage items (time to store the hand luggage (Schultz, 2018a)) and arrival time at assigned aircraft door. Furthermore, several process characteristics could be recorded during simulation runs, such as waiting times, number of interactions, and current cabin status. To create the time needed to store the hand luggage, a triangular distribution provides a stochastic time value depending on the number of items (see Schultz et al., 2008, 2013b). The agents are sorted with regard to the current boarding strategy (depending on their seats). From this sequence, a given percentage of agents are taken out of the sequence (conformance rate) and inserted into a position which contradicts the current strategy (e.g. inserted into a different boarding block). According to arrival time distribution and boarding sequence, each agent receives a timestamp, where the agent enters the simulation and appears at the aircraft door queue. When the simulation starts, the first agent of this queue enters the aircraft by moving from the queue to the entry cell of the aisle grid (aircraft door), if this cell is free. In each simulation step, all agents located in the row are moved to the next cell, if possible (free cell and not arrived at the seat row), using a shuffled sequential update procedure (emulate parallel update behavior, cf. Schultz, 2014). If the agent arrives at his seat row, he waits at this position for as long as it takes to store the hand luggage. Depending on the seat row condition (e.g. assigned window seat but blocked aisle or middle seat or both are blocked), an additional waiting time is stochastically generated to perform the seat shuffle. During the whole waiting process, no other agent can pass in. If the agent's waiting process finishes, he is set to his seat and the aisle cell is set free. The consequences of this seat shuffle are shown in Fig. 4, where the passengers are color-coded with green (no seat interactions), yellow (minor interactions), orange (medium interactions), and red (high interactions). Each boarding scenario is simulated 100,000 times to achieve reliable calculation results in order to determine the average boarding time with corresponding standard deviation (measure of stability). In close cooperation with German airports and airlines, field trials were conducted to calibrate input parameters of the stochastic model and to validate the simulation results. Finally, recorded boarding times from the field and boarding times provided by the simulation environment show deviations smaller than 5% (Schultz, 2018a).

2.3. Evaluation of boarding progress

The aircraft boarding time possesses a stochastic characteristic, which results in a statistical distribution of the boarding time (cf.

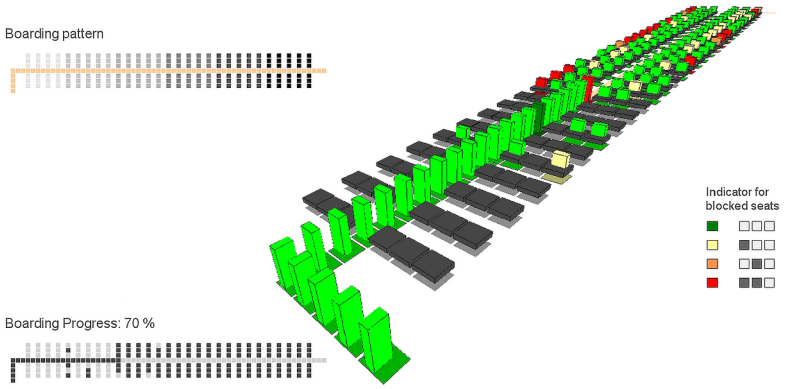

Fig. 4. Visualization environment for the aircraft boarding progress considering a *back-to-front* strategy. Interactions of passengers during seating are emphasized with a color scale from green (no interactions) to red (high interactions), by means of no blocked seats/empty row (green) and two blocked seats (red) when passenger wants to take the assigned seat. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

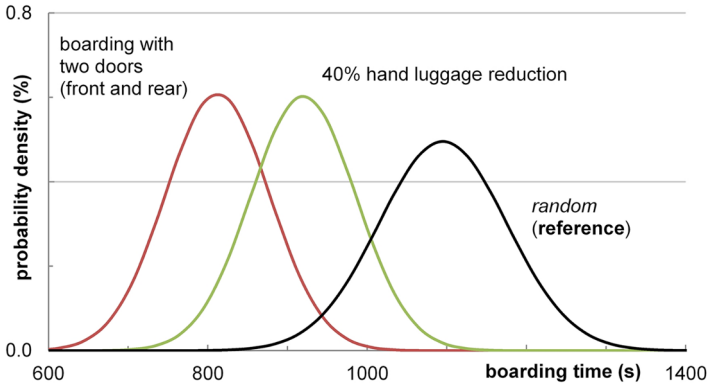

Fig. 5. Distribution of simulated boarding times using two different boarding scenarios compared against the reference scenario (*random* boarding, cf. Schultz (2018b,c)).

Schultz, 2018b). In Fig. 5, three different boarding scenarios are shown with their probability density to demonstrate the effect on boarding time with reduction of hand luggage items and boarding with two doors against a reference boarding time distribution (*random* boarding). The effect on the corresponding boarding time is twofold: both scenarios possess a smaller standard deviation (more stable) and are faster than the reference boarding scenario.

However, to predict the end of a specific boarding event, a combination of expectation concerning the future progress and the already reached progress is required. In the course of aircraft boarding, passengers will reach their seats and have to be no longer considered for the (stochastic) prediction of boarding time t_b. Thus, the prediction could be divided into a fixed, realized boarding time t_r and a versatile, stochastic component t_s of the boarding time, to cover still-existing uncertainties. During the course of boarding, the number of passengers seated in the aircraft increases constantly and the stochastic component of the boarding time has a decreasing influence on the final boarding time. Fig. 6 demonstrates the implication of decreasing uncertainty during the course of boarding, indicated by higher values for the corresponding probability density.

To provide a closer view of the uncertainties during the course of boarding, an example is given in Fig. 7, with a boarding progress of 90% and 13 free seats. From a general, qualified point of view, a free seat at the window with two already occupied seats will result in a high interference between passengers (seat shuffle: two passengers have to move into the aisle), whereas free aisle seats demand no interference during seating. This effect could be more prominent if other passengers have to wait during the luggage/seating process. Having this in mind, the free seats could be ranked according to their interference potential. In Fig. 7, this interference is qualitatively clustered by high, medium, and low level of interference. Depending on the current boarding sequence, this potential interference could result in high numbers of interactions if the seats with the highest potentials are boarded first, or result in small numbers if the seats are boarded vice versa.

This exemplary boarding sequence emphasizes that a reliable predication of the boarding demands both current boarding progress and the evaluation of upcoming interactions between passengers. In the next section, a metric is briefly introduced which comprehensively considers expected passenger interactions during boarding. This metric (of complexity) depends on the already realized boarding progress and three major indicators: seat position in seat row, seat row position in cabin, and boarding sequence.

2.4. Complexity metric

The interactions caused by the seat position in the seat row are quantified by positions of the specific seats (window, middle, aisle)

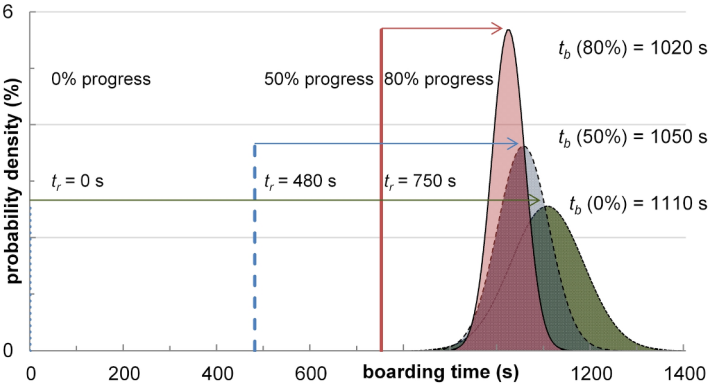

Fig. 6. Boarding time distribution during the aircraft boarding based on a realized boarding progress of 0%, 50% and 80%.

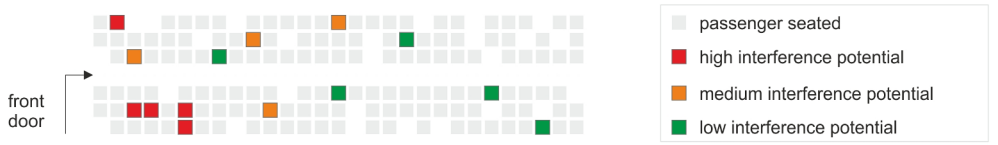

Fig. 7. 90% boarding progress with 13 free seats. Depending on the seat coordinates (row and position), a qualitative level of interference could be derived (see colour-coded potential ratings in the chart key). Passengers using a seat in the back or at the aisle will only have minor potential for interference.

and corresponding operational seat conditions C_{seat}: *not booked* ($C_{seat} = 0$), *free* ($C_{seat} = 1$), and *occupied* ($C_{seat} = 2$). Since the A320 reference is a single-aisle aircraft with three seats on the left and the right sides respectively, three seats are aggregated to a seat row condition $C_{row,\ left|right}$. A trinary aggregation of seat conditions per (half) row, according to Eq. (1), results in 27 distinct values for $C_{row,\ left|right}$. In the general formulation of Eq. (1), c represents the number of different seat conditions; in the current case $c = 3$.

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

As an example, if all seats in a row are occupied, C_{row} is 26 ($= 3^0 \times \mathbf{2} + 3^1 \times \mathbf{2} + 3^2 \times \mathbf{2}$); if aisle seat is occupied, middle seat is not booked, and window seat is free, C_{row} is 11 ($= 3^0 \times \mathbf{2} + 3^1 \times \mathbf{0} + 3^2 \times \mathbf{1}$). Each C_{row} condition results in a specific number of interactions between passengers and depends on the seat position of the arriving passenger. If all seats are free, there will be no interactions at all. If, however, the aisle seat is occupied, the next arriving passenger demands a seat shuffle with a minimum of four movements (cf. 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. In the following, 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). Fig. 8 demonstrates the transition between different seat row conditions 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. If the boarding sequence starts with the middle seat and ends with the window seat, eleven passenger movements are necessary, and P_r develops: $P_{r,\ step\ 0} = 8.3$, $P_{r,\ step\ 1} = 8.0$, and $P_{r,\ step\ 2} = 9$.

The negative effect of high P_r values increases (longer waiting, boarding time) if other passengers are hindered in reaching their corresponding seat row and queue in the aisle. This effect depends on the position of the row where the interaction takes place and 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. Here, only passing passengers with higher seat row numbers are considered. This is done by counting all seats which are currently *free* ($C_{seat} = 1$) in the concerned row and the rows behind, whereby this number is stored as the value $p_{b,r}$ (number of passengers located in and behind this row r). To weight the influence of $p_{b,r}$ an exponential function with A as scaling factor is chosen. In Eq. (2), P_r is weighted in order to consider the indirect interference caused by aisle queues and summed up to an aircraft-wide interference potential P, where m represents the maximum number of rows.

$$P = \sum_{r=1}^{m} P_{r,\text{left|right}}\, e^{A p_{b,r}} \tag{2}$$

The aircraft-wide interference potential P is based on the expected value of P_r, since several future seat states are possible. To provide a more comprehensive dataset, the difference between maximum and minimum value of P_r is calculated as well (ΔP), as measurement of convergence. It is expected that lower ΔP (less spread) will result in a more reliable prediction of the boarding time.

To complement the complexity metric, the quality of the passenger sequence k has to be considered as well (Eq. (3)). Since the passengers seated are already covered in P, the boarding sequence contains passengers who have already passed the boarding counter

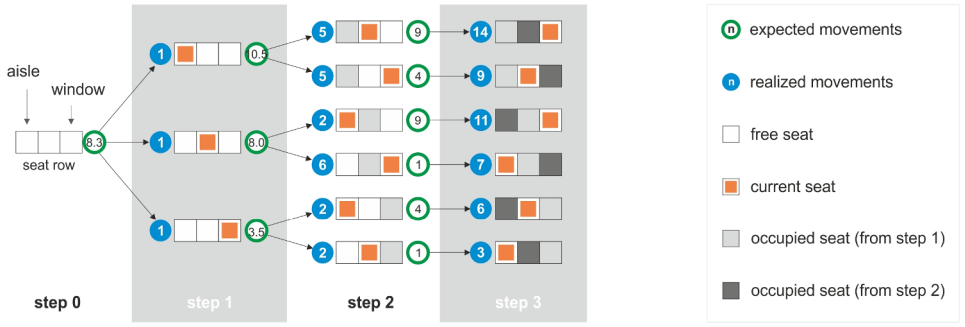

Fig. 8. Progress of seat row development (aisle seat right and window seat left) indicating both the movements/interactions expected (green circle) and realized (blue circle). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

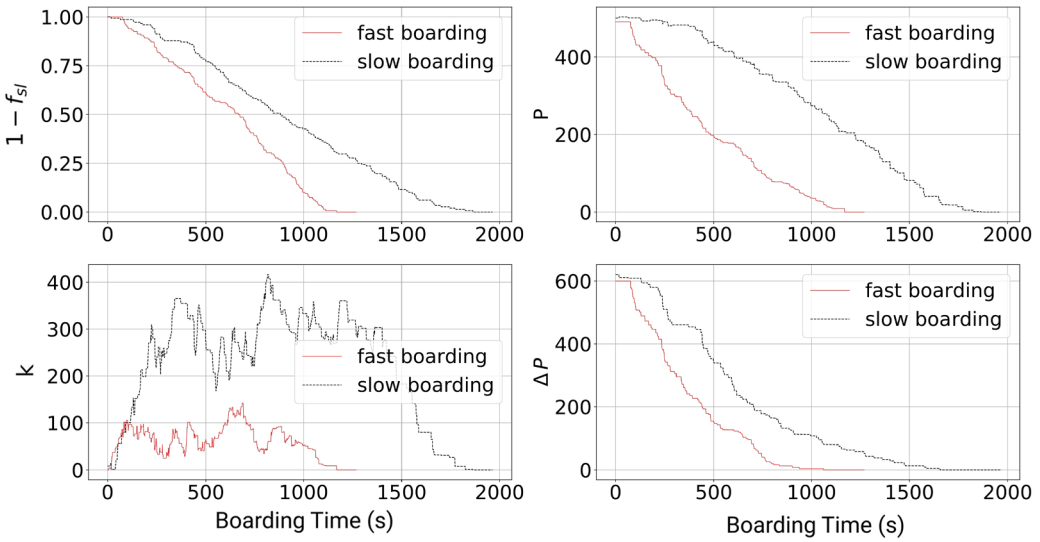

Fig. 9. Complexity metric to evaluate the boarding process and progress: seat load (top left), interference potential (top right), boarding sequence (bottom left), and convergence of interference potential (bottom right).

but not been seated in the aircraft in the chronological order of the arrival at the boarding counter. Beginning with the first passenger i in the queue (with n members), each subsequently following passenger j is marked as influenced ($d = 1$) if he wants to reach a seat row with a higher number than the passenger in front of him. In Eq. (3), an exponential function is used to consider the weighted distance of the passengers in the queue, and the expected seat row interference potential P_{ri} for passenger i is also taken into account. While iterating over all passengers in the queue, the current passenger i will be virtually seated, which results in a continuous update of future seat row states for the subsequently following passengers. A lower value of k indicates a lower level of influence between the passengers in the queue.

$$k(t) = \sum_{i=1}^{n} \sum_{j=i}^{n} d_{i,j} e^{B(i-j)} P_{ri,\,\text{left|right}}$$
$$d_{i,j} = \begin{cases} 1, & \text{seat row}_{paxi} \leq \text{seat row}_{paxj} \\ 0, & \text{seat row}_{paxi} > \text{seat row}_{paxj} \end{cases} \tag{3}$$

Finally, the complexity metric consists of four different indicators to evaluate the boarding progress. These indicators address the major, time-dependent drivers of boarding: progress of passengers seated $1 - f_{sl}$ (seat load), interference potential P (different blocking conditions), convergence of interference potential ΔP (difference of maximum and minimum values), and passenger sequence k (boarded, but not seated). We used the $1 - f_{sl}$ definition for seat load function, to provide a consolidated picture of the complexity metric; all indicators end with zero when the boarding finishes. Fig. 9 demonstrates the characteristic of the developed complexity metric using a fast and slow boarding event exemplarily. The fast boarding event depicts a lower interference potential P, quicker convergence ΔP, smaller values for the passenger sequence k and, finally, a faster boarding progress ($1 - f_{sl}$).

3. Machine learning

To overcome resulting complexities and dynamic effects of passenger boarding, time predictions can be applied using advanced statistical procedures, such as neural networks. Neural networks are a family of network types of machine learning (ML) with the ability to learn and adjust. The setup of a neural network depends on the specific task. Training it with certain datasets will adjust the parameters and transfer the input values to a learned output set of data. In the field of ML, there are several opportunities for statistical analyses, sequence predictions and knowledge approximations. For value-discrete time-series prediction, Multi-Layer Perceptron (MLP) and Recurrent Neural Network (RNN) approaches are mainly established and offer different subtypes. Furthermore, Time Delay Neural Networks (TDNN) or Radial Basis Function (RBF) networks would also be suitable for the objectives in the given task of boarding time prediction. To capture the core idea of identifying long-term dependencies and correlations between the indicators of the complexity metric, RNN and TDNN are the most suitable approaches, whereby RNN plays a much more important role in the context of sequence prediction. For simple problems, ML methods can be a too-powerful tool, since the effort to prepare the model is significantly higher than the effort needed to solve the problem. In these cases, Gaussian processes and ARIMA (Auto Regressive Integrated Moving Average) models provide alternative approaches which are not, however, discussed in this paper, as they are not able to represent a suitable model for the given task of real-time prediction based on nonlinear time series. This section provides an overview concerning data transformation techniques to convert boarding time series to applicable datasets for machine learning (cf. Yu et al., 2006), followed by a brief description of neural network structures and introduction of LSTM models. The section closes with an overview about specific learning algorithms for recurrent neural network types.

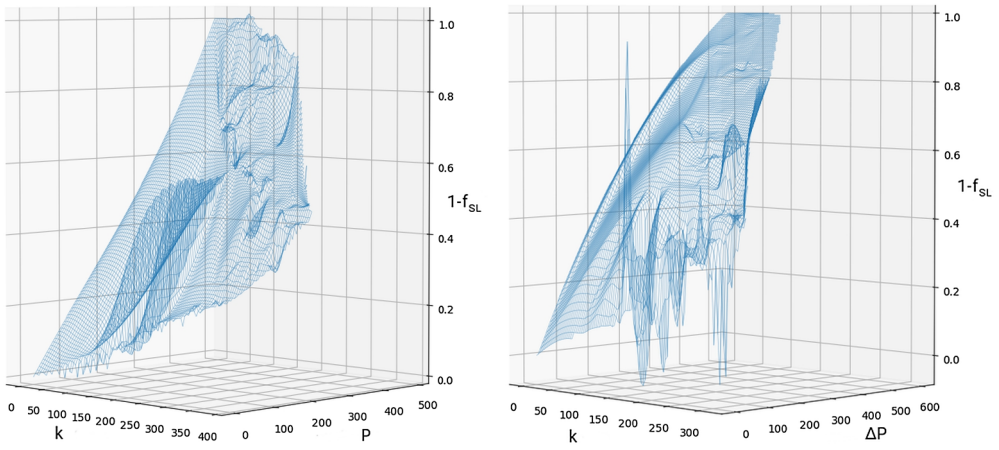

Fig. 10. Comparison of a three-dimensional relationship representation: (left) quality of passenger queue (k), interference potential (P) and seat load ($1 - f_{sl}$) and (right) quality of passenger queue (k), variability of interference potential (ΔP) and seat load ($1 - f_{sl}$). The relationships between interference and variability are similar in data dynamics (representation of one exemplary *random* boarding simulation).

3.1. Dataset and feature selection

As output of the boarding simulation, we use 25,000 datasets of independent simulation runs for each boarding strategy (*random*, *block*, *back-to-front*, *outside-in*, *reverse pyramid*, *individual*). This results in a total number of 150.000 available datasets. Each dataset contains the four mentioned features: P, ΔP, $1 - f_{sl}$ and k. Considering the real applications, we want to keep the number of features low. For this reason, we chose a uni-variate and multi-variate approach. The feature set of the uni-variate approach contains the feature $\{1 - f_{sl}\}$, the set of the multi-variate approach contains $\{1 - f_{sl}, P\}$. According to the complexity metric (Section 2.4), the indicator ΔP is directly related to P and redundancy is to be expected between ΔP and P. Because P is a direct indicator of the quality of the dynamic boarding progress, this feature is preferred to the feature k. Fig. 10 confirms this assumption.

Table 1 contains a descriptive view of the given dataset, clustered by the boarding strategies and the 4 indicators of the complexity metric. The characteristics are average value (mean), standard deviation (std), minimum (min) and maximum (max), as well as the 25th, 50th, and 75th quantiles. The table reflects that the *individual*, *reverse pyramid*, and *outside-in* strategies result in a fast boarding progress, while *back-to-front*, *block*, and *random* boarding are slower on average (cf. Schultz, 2018b). *Random* boarding is one of the slower methods, which makes the analysis interesting as to whether this slower *random* boarding can be used for training to predict faster procedures (such as *individual*). Since *random* does not follow any restrictions, the dynamic passenger behavior should nevertheless be able to be mapped quickly. Table 1 is intended to serve as an estimate of the underlying database.

3.2. 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, such as the interference potential P, which will be exemplarily used to introduce supervised learning. In a straightforward implementation, the next value of the interference potential (output) only depends on the current (input) value $P(t) \rightarrow P(t + 1)$. To derive predictions for long-term sequences with a time horizon d (interval value from given t_0 measured in discrete time steps), the values need to be iterated from $t_{0\text{-}d}$ to t_0, where the predicted value of every time step receives an input for the next step. This sliding window is shown in Table 2 with a time horizon $d = 2$. Table 2 additionally demonstrates that a prediction with three input parameters could initially start when all input parameters are set ($t = 2$). Furthermore, this example shows that least nominal scaled data are needed, and it is expected that the accuracy of the prediction increases if a longer time horizon is taken into account (history). The consideration of historic values results in a sliding window effect, which is fundamental to prepare a discrete time series as input for a supervised learning approach. The size of this sliding window will be adjustable and has a significant influence on prediction, especially on the computational time and interdependencies (e.g. time-delayed correlations). In this context, the sliding window emphasizes the opportunity to handle a particular time series, but time series with more than one observation at each time step could be handled as well.

Traditionally, time series are differentiated into uni-variate (observation of one single variable) and multi-variate (observation of datasets) series. In the case of aircraft boarding, the multi-variate time series consists of interference potential P, seat load $1 - f_{sl}$, passenger sequence k, and convergence of interference potential ΔP. The application of the boarding model provides values for every single indicator, which could be used as multi-layer input considering current and past values. Since classical statistical methods

Table 1
Descriptive statistics of all 4 features – passenger queue (k), interference potential (P), variability of interference potential (ΔP) and progress of seat load ($1 - f_{sl}$) for the six concerned boarding strategies covering 25.000 simulation runs each.

random	mean	std	min	25th	50th	75th	max
k	186.78	94.61	0	118.73	199.9	259.76	362.89
P	255.12	155.93	0	115.74	259.03	396.6	495.23
ΔP	183.77	189.98	0	19.76	110.54	313.39	608.23
$1 - f_{sl}$	0.47	0.31	0	0.19	0.45	0.74	1.00
back-to-front	mean	std	min	25th	50th	75th	max
k	232.6	128.48	0	134.56	235.41	336.93	472.05
P	242.71	157.57	0	101.17	231.06	386.04	494.57
ΔP	227.67	195.23	0	49.86	180.92	384.69	608.34
$1 - f_{sl}$	0.48	0.32	0	0.19	0.46	0.77	1.00
block	mean	std	min	25th	50th	75th	max
k	222.62	141.43	0	124.75	200.12	309.83	544.59
P	254.73	152.67	0	127.46	250.08	388.99	494.66
ΔP	239.78	189.89	0	57.37	229.89	374.15	608.46
$1 - f_{sl}$	0.5	0.31	0	0.23	0.52	0.76	1.00
outside-in	mean	std	min	25th	50th	75th	max
k	110.29	62.89	0	63.88	108.40	155.22	254.30
P	196.84	154.66	0	59.84	161.30	321.40	493.77
ΔP	182.28	197.42	0	12.05	100.24	322.10	607.50
$1 - f_{sl}$	0.51	0.31	0	0.23	0.51	0.79	1.00
reverse pyramid	mean	std	min	25th	50th	75th	max
k	101.69	57.94	0	58.13	101.61	144.17	232.16
P	214.74	158.36	0	68.23	194.67	346.75	493.99
ΔP	220.69	201.24	0	32.82	167.29	378.02	608.38
$1 - f_{sl}$	0.51	0.32	0	0.23	0.50	0.80	1.00
individual	mean	std	min	25th	50th	75th	max
k	63.99	38.55	0	35.08	59.45	88.61	168.09
P	187.92	153.05	0	54.42	148.22	306.09	493.87
ΔP	175.11	197.56	0	8.43	90.51	308.43	608.12
$1 - f_{sl}$	0.51	0.31	0	0.23	0.51	0.78	1.00

Table 2
Transformation of input data (sliding window).

Time t	Input 1 X(t − 2)	Input 2 X(t − 1)	Input 3 X(t)	Output Y(t + 1)
0	–	–	*P(0)*	*P(1)*
1	–	*P(0)*	*P(1)*	*P(2)*
2	*P(0)*	*P(1)*	*P(2)*	*P(3)*
3	*P(1)*	*P(2)*	*P(3)*	*P(4)*

either work insufficiently in rebuilding complex functions (multi-variate approaches are mostly limited to linear approximations) or are not able to represent a wide range of possible underlying processes, the application of intelligent machine learning approaches is favored (cf. Karlaftis and Vlahogianni, 2011). As an example, the boarding dataset could be transformed to a supervised learning problem using P and $1 - f_{sl}$ with a window size of one, as a multi-to-one time series (three input features and one output value, see

Table 3
Multi-to-one time series using interference potential and seat load.

Time t	Input 1 X_1	Input 2 X_2	Input 3 X_3	Output 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

Table 4
Neural network adjustments.

Input 1 X_1	Input 2 X_2	Input 3 X_3	Output Y	*Error*
$P(0)$	–	$P(1)$	$1 - f_{sl,\ pred}\ (1)$	$1 - f_{sl,\ pred}\ (1) \leftrightarrow 1 - f_{sl,\ real}(1)$
Neural Network Parameter Adjustment (*Learning*)				
$P(1)$	$1 - f_{sl,\ pred}\ (1)$	$P(2)$	$1 - f_{sl,\ pred}\ (2)$	$1 - f_{sl,\ pred}\ (2) \leftrightarrow 1 - f_{sl,\ real}\ (2)$
Neural Network Parameter Adjustment (*Learning*)				
$P(2)$	$1 - f_{sl,\ pred}\ (2)$	$P(3)$	$1 - f_{sl,\ pred}\ (3)$	$1 - f_{sl,\ pred}\ (3) \leftrightarrow 1 - f_{sl,\ real}\ (3)$
Neural Network Parameter Adjustment (*Learning*)				
$P(3)$	$1 - f_{sl,\ pred}\ (3)$	$P(4)$	–	

Table 3) (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). To do so, the predicted value of a sequence (for example $1 - f_{sl}$) is added to the upcoming input sequence (see Table 4). The estimated value $1 - f_{sl,\ real}$ is only used for deviation analysis of the network's output compared to the real output. Using this online dataset adjustment, 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.

3.3. Neural network models

A standard recurrent neural network (RNN) computes the hidden vector sequence $h = (h(1), \ldots, h(T))$ and output vector sequence $y = (y(1), \ldots, y(T))$ for a given input sequence $x = (x(1), \ldots, x(T))$ as shown in (4) and (5):

$$h(t) = H(W_{hy}x(t) + W_{hh}h(t-1) + b_h) \tag{4}$$

$$y(t) = W_{hy}h(t) + b_y. \tag{5}$$

The looped structure of RNN allows information to be passed from one step to the next. Fig. 11 depicts the unfolded loop of an RNN model with module A being the simplified inner structure. The self-connected model (left) is split into multiple copies of the same network (right) considering, however, different inputs, which are related to the time.

Regarding (4) and (5), the W terms denote weight matrices 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. Unlike classical feedforward neural networks, RNN models perform well in detecting and processing long-term correlations of inputs and outputs. Fig. 12 illustrates an effect of the input vectors $x(0)$ and $x(1)$ on the hidden output $h(t + 1)$. Such interdependencies, delayed over certain time steps, are an essential part of a multi-variate data analysis, as the dynamic of the system needs to be described through the influence of indicators on 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. They also have a chain-like structure, but the module A has a different composition. Instead of having a single neural network layer, the inner workings of LSTM modules are split into four gates. The LSTM structure is implemented through the following Eqs. (6)–(9):

$$i(t) = \sigma(W_{xi}x(t) + W_{hi}h(t-1) + W_{ci}c(t-1) + b_i) \tag{6}$$

$$f(t) = \sigma(W_{x}fx(t) + W_{h}fh(t-1) + W_{c}fc(t-1) + b_f) \tag{7}$$

$$c(t) = f(t) * c(t-1) + i(t) * \tanh(W_{xc}x(t) + W_{hc}h(t-1) + b_c) \tag{8}$$

$$o(t) = \sigma(W_{xo}x(t) + W_{ho}h(t-1) + W_{co}c(t) + b_o) \tag{9}$$

σ and *tanh* represent the specific, element-wise applied activation functions of the LSTM. i, f, o and c denote the mentioned inner-cell

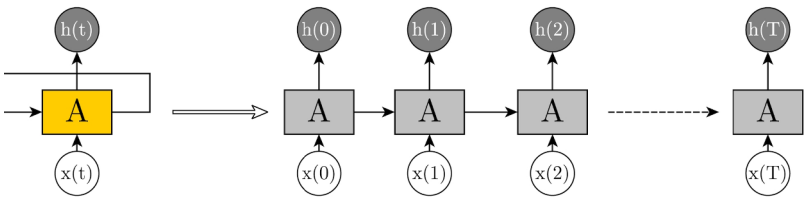

Fig. 11. Unfolded RNN structure for computations from 0 to T.

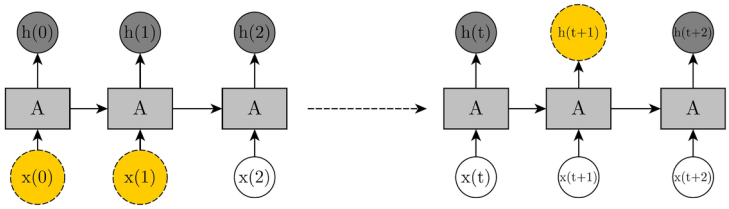

Fig. 12. Long-term interdependencies in RNN models encompassing a connection of the input vectors *x(0)* and *x(1)* to the hidden output vector *h (t + 1)*.

gates, respectively the input gate, forget gate, output gate, and cell activation vectors. Function *c* needs to be equal to the hidden vector *h*. The *W* terms again denote weight matrices.

As LSTM models are RNN-based models, the overall composition is similar, so a set of inputs is computed to an output set through a module *A* (see to Figs. 11 and 12). Unlike basic RNN models, *A* is more complex in this case: Fig. 13 depicts the availability of the cell memory, which can be interpreted as the information store, and the inner workings of the described four gates: computing the input, updating the cell, forgetting about certain information and setting the output (cf. Gers et al., 2000).

To train RNN structures, a backpropagation through time algorithm is applied (cf. 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 neural network can adjust its parametrization to retrieve given target values. Normally, RNN structures have problems with a vanishing gradient (a backpropagation of *tanh* values near 0 or 1 over time might lead to a blowing-up or disappearing). LSTM networks address this problem by splitting in 4 inner-cell gates with the opportunity to "store" knowledge in the inner-cell gate. RNN-based models are characterized by the number of hidden layers ($n_{hiddenlayer}$), 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).

3.4. 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. In Fig. 14, the network structure is presented for both cases one-to-one (solid) and multi-to-multi regression (dotted).

Each layer consists of specific tasks in the framework. The LSTM (recurrent) layer represents the specific ML model. The input layer is a *Keras* tensor object from the underlying backend (TensorFlow) and contains certain attributes for building a *Keras* model, just by knowing only inputs and outputs of the model. 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 = activation(dot(input, kernel) + bias),* where *activation* is the element-wise activation function passed as the *activation* argument, *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).

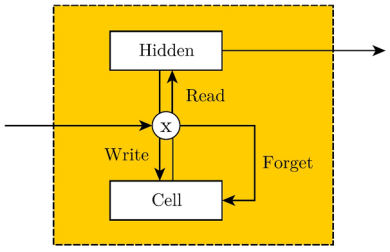

Fig. 13. Inner structure of an LSTM module *A* emphasizes hidden gates and the cell information memory.

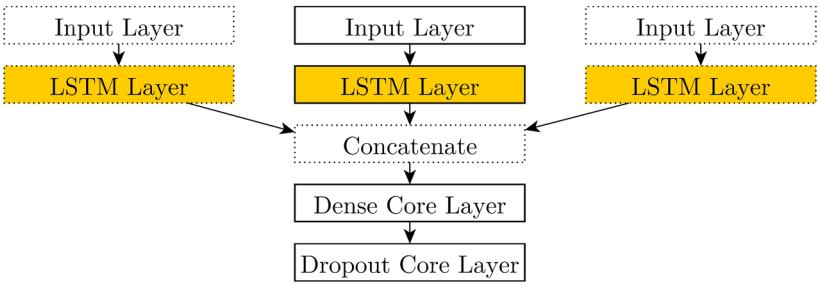

Fig. 14. Layer design of simulation framework.

4. Simulation framework and application

The input data for the LSTM model is provided by the stochastic boarding model, which covers several boarding strategies, such as *random*, *block*, *back-to-front*, *outside-in*, or *individual* boarding (cf. Schultz, 2018b). 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*) using CUDA as parallel computing platform and application programming interface.

4.1. Scenario definition

Table 5 provides an overview of the analyzed scenarios (A, B, C), focusing on the prediction of boarding progress of *random* (passengers have pre-assigned seats and enter the aircraft unconstrained, with no specific order) and *individual* boarding (each passenger has a specific position in the waiting queue at the gate, cf. Steffen (2008)) of A320 aircraft. Here, *random* boarding is used as a common process and *individual* boarding as one of the fastest boarding processes. In scenario A, the prediction of *random* boarding is based on learning of *random* boarding scenarios. In scenario B, the learned interrelationships from *random* boarding are used to predict the boarding progress of *individual* boarding. In scenario C, the progress of *individual* boarding is predicted on the basis of all available boarding datasets (cf. Schultz, 2018b). As input values P, $1 - f_{sl}$, or both are used to predict the output $1 - f_{sl}$. All three scenarios are computed for a given set of start times: 300 s, 400 s, and 600 s after boarding starts.

The supervised learning demands for a separation of the 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. For the given tasks, we use 1000 randomly chosen boarding events to train the LSTM and 100 randomly chosen non-compiled/non-trained boarding progresses to evaluate the prediction.

4.2. Results

We trained the model within 27 different variants: three 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). As we have two one-dimensional input sets ($1 - f_{sl}$ and P), a more complex structure is needed, so we set $n_{hiddenlayer} = 120$ (number of hidden LSTM layer next to input and output layer) for these variants. In the case of uni-variate prediction, we set $n_{hiddenlayer} = 40$. A higher number of $n_{hiddenlayer}$ increases the ability of the network to understand complex data structures, but also increases the complexity of learning. 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 in 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 and 2.0% for multi-dimensional applications. Fig. 15 depicts the prediction results, using a slow and fast boarding progress. Here, the prediction starts at 400 s. The comparison of real and predicted boarding progress demonstrates that the LSTM

Table 5
Scenario definition (see Schultz (2018b) for further details about different boarding strategies).

Scenario	boarding strategy learned	boarding strategy predicted	input from complexity metric	prediction start time t (s)
A1	*random*	*random*	$\{P, 1 - f_{sl}, [P, 1 - f_{sl}]\}$	{300, 400, 500}
A2	*individual*	*individual*		
B	*random*	*individual*		
C	all (*random*, *block*, *back-to-front*, *outside-in*, *reverse pyramid*, *individual*)	*individual*		

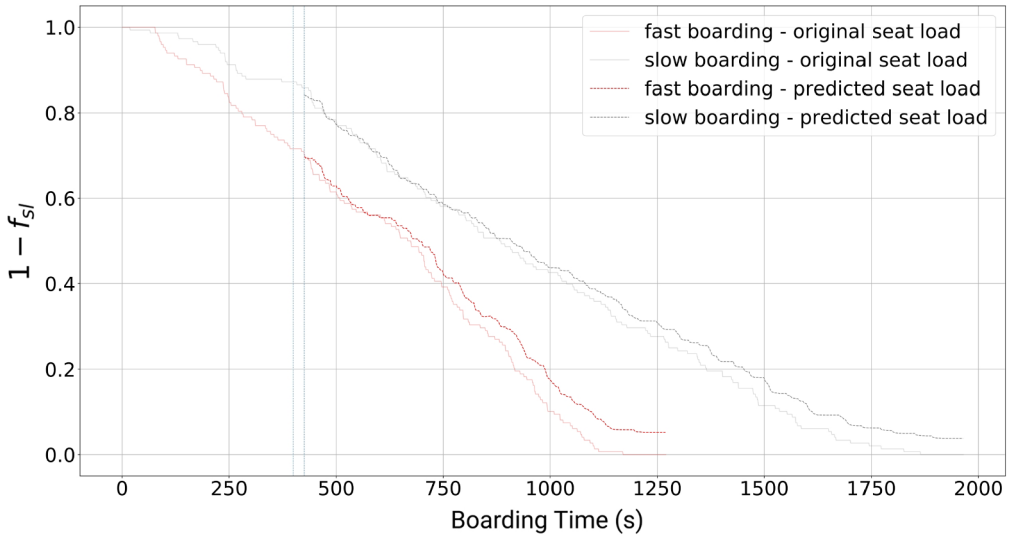

Fig. 15. Prediction of seat load progress with LSTM model for two exemplary processes (start at 400 s).

model is a promising candidate for boarding time prediction.

The following Table 6 provides an overview of the accuracy of each single model to the corresponding scenario including the computation time. The accuracy is the mean error value of 100 tested scenarios after training phase (1000 sample sets). In Table 6, cells are color-coded, whereby 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, the neural network models receive the knowledge to end at point 0 (end of boarding) in every case of training. An insufficiently trained model might show bifurcations

Table 6
Results of computations for the scenarios A, B and C considering time series accuracy, accuracy of final boarding time and computational time *t*. Bold values represent best values of specific input configuration.

scenario	input (uni-variate)	input (uni-variate)	input (multi-variate)
A1	$[1-f_{sl}]$	$[P]$	$[1-f_{sl}, P]$
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
A2	$[1-f_{sl}]$	$[P]$	$[1-f_{sl}, P]$
300	NaN t = 1h 21min	NaN t = 1h 29min	210.1s t =2h 17min
400	412.8s t = 0h 49min	NaN t = 1h 19min	92.9s t = 2h 02min
500	**299.4s** t = 0h 47min	**503.0s** t = 0h 55min	**87.1s** t = 1h 57min
B	$[1-f_{sl}]$	$[P]$	$[1-f_{sl}, P]$
300	NaN t = 1h 19min	NaN t = 1h 19min	243.0s t = 2h 01min
400	**197.9s** t = 1h 21min	394.7s t = 1h 00min	**74.1s** t = 2h 16min
500	282.2s t = 0h 55min	**391.1s** t = 1h 03min	79.8s t = 2h 20min
C	$[1-f_{sl}]$	$[P]$	$[1-f_{sl}, P]$
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

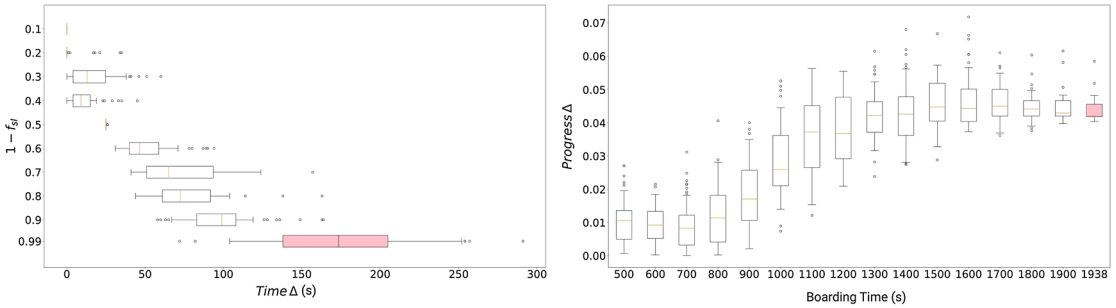

Fig. 16. Scenario A1 (*random* - *random*) - time (left) and progress (right) differences after 1000 training iterations and 100 tests.

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 and only the multi-variate input results in reliable output values. Uni-variate models might tend to adequately predict the final boarding time (see $1\text{-}f_{sl}$ at t = 500 s, scenario A2) but possess a non-nominal process identification.

Figs. 16–19 show the time and progress differences, using 400 s (start time for prediction) as a reference start value. The box-plots of the time differences refer to the values illustrated in Table 6, using an interquartile ratio of 0.7. The red-colored boxes include the last computed steps of the simulations (99%).

Scenario A is shown in Fig. 16 (A1, *random*, slow) and Fig. 17 (A2, *individual*, fast). The left pictures show the time differences: the *individual* boarding does not differ until 40% boarding progress. Since time variation in *random* boarding is significantly higher than in *individual* boarding, the inner progress differences are also higher (pictures on the right). Both prediction results point out a positive offset over time.

The major difference between scenario A and scenarios B and C is that in scenario A the LSTM is trained with appropriate datasets. In scenario B, the *individual* boarding is predicted with *random* datasets (explicitly not containing *individual* boarding) and scenario C uses all available datasets for training (only excluding datasets for testing). In doing so, the computational time for training increases significantly (see Table 6). Due to the higher bandwidth of operational progresses in scenarios B and C, the LSTM model is able to identify more interdependencies during the training process, which results in a more precise prediction (see Figs. 18 and 19).

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. We set a specific focus on the aircraft ground operations, which provide operational milestones for both the current and the destination airport. These mutual interdependencies between airports result in system-wide, far-reaching effects (reactionary delays). The ground trajectory of an aircraft primarily consists of the handling processes at the stand (deboarding, catering, fueling, cleaning, boarding, unloading, and loading), which are defined as the aircraft turnaround. To provide a reliable prediction of the turnaround, the critical path of processes has to be managed in a sustainable manner. The turnaround processes are mainly controlled by the ground handling and airport or airline staff, except the aircraft boarding, which is driven by the passengers' experience, willingness or ability to follow the proposed procedures.

This paper provides, for the first time, an approach to predict aircraft boarding progress, which is expected to add significant benefit to future aircraft ground operations (e.g. precise scheduling due to reduction of operational uncertainties). Our approach shows how sensor information from a connected cabin could be used to derive the progress of the passenger boarding. Since today there are no operational sensor data available from inside the cabin, we simulated the boarding process with a validated boarding model (Schultz, 2018a,b). This model provides detailed data about passenger movements and operational processes (e.g. waiting in

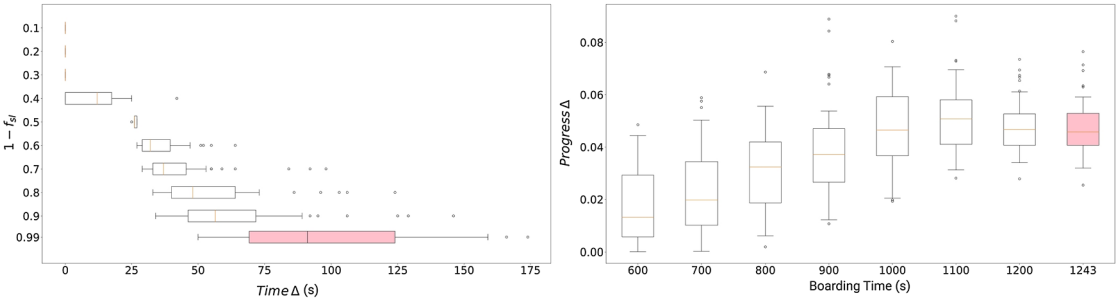

Fig. 17. Scenario A2 (*individual* - *individual*) - time (left) and progress (right) differences after 1000 training iterations and 100 tests.

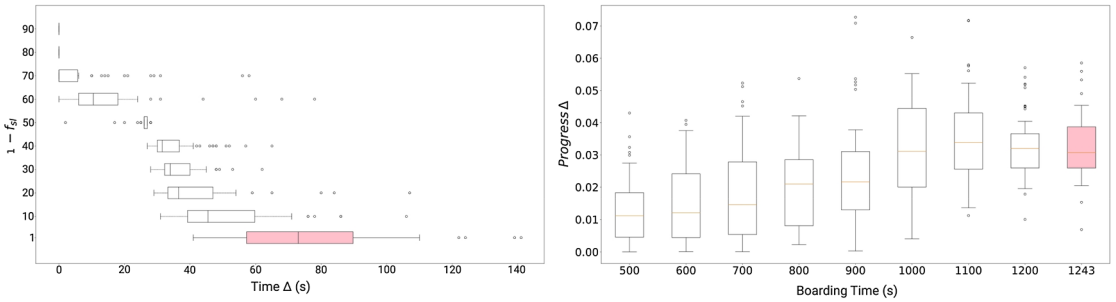

Fig. 18. Scenario B (*random - individual*) - time (left) and progress (right) differences after 1000 training iterations and 100 tests.

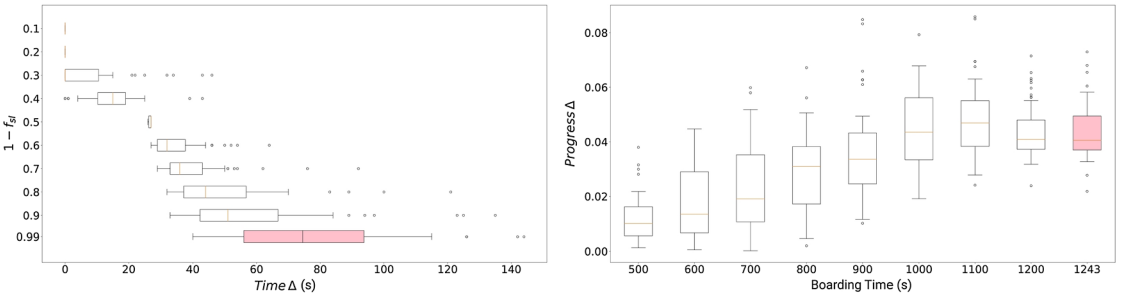

Fig. 19. Scenario C (*all - individual*) - time (left) and progress (right) differences after 1000 training iterations and 100 tests.

the aisle, storage of hand luggage, or used seats). In particular, we aggregated these data to complexity values, which describe the expected interactions of passengers comprehensively (2018c). The complexity metric is a multi-variate input for an LSTM model, which is trained with boarding simulation data (time series) and enables a prediction of the final boarding time. We could demonstrate that the proposed complexity metric is a necessary element for prediction of the aircraft boarding progress. Whilst the progress of the seat load was insufficient to reliably predict the boarding (high deviations), together with the complexity metric, the deviations are reduced by up to 75% (cf. Table 6). A closer look at differences between boarding progress and prediction shows an inherent positive offset. If this offset could be classified as constant, the prediction showed only a difference of ± 20 s between prediction and underlying boarding simulation. Therefore, we see our LSTM approach model as a promising candidate for extended investigations. Furthermore, a hardware prototype of a connected cabin (seat and floor sensors) proves the operational feasibility of our prediction concept (cf. Schultz, 2018d).

As a next step, we want to focus on the robustness of the LSTM prediction model and quality of the input factors. Thus, we want to transform the given RNN models to a feedback control structure representing a dynamical system. In doing so, we will be able to apply several studies regarding robustness and dynamic behavior, such as Lyapunov stability (Reitmann and Schultz, 2017). First tests with the LSTM input indicate that a smoothed input stream results in better prediction results (Reitmann and Schultz, 2018).

References

Airbus, 2017. Airbus A320 Aircraft Characteristics - Airport and Maintenance Planning.
Bachmat, E., Berend, D., Sapir, L., Skiena, S., Stolyarov, N., 2009. Analysis of aircraft boarding times. Oper. Res. 57, 499–513.
Bazargan, M., 2007. A linear programming approach for aircraft boarding strategy. Eur. J. Oper. Res. 183 (1), 394–411.
Bronsvoort, J., McDonald, G., Porteous, R., Gutt, E., 2009. Study of aircraft derived temporal prediction accuracy using FANS. In: 13th Air Transport Research Society (ATRS) World Conference.
Cao, X., Zhu, X., Tian, Z., Chen, J., Wu, D., Du, W., 2018. A knowledge-transfer-based learning framework for airspace operation complexity evaluation. Transport. Res. Part C: Emerg. Technol. 95, 61–81.
Chung, Ch.-A., 2012. Simulation design approach for the selection of alternative commercial passenger aircraft seating configurations. J. Aviation Technol. Eng. 2, 100–104.
Cook, A.J., Tanner, G., 2015. European Airline Delay Cost Reference Values. Technical Report. EUROCONTROL Performance Review Unit, Brussels.
Du, J.Y., Brunner, J.O., Kolisch, R., 2014. Planning towing processes at airports more efficiently. Transp. Res. Part E 70, 293–304.
Duchi, J., Hazan, E., Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. JMLR 12, 2121–2159.
Eurocontrol, 2017. Performance Review Report – An Assessment of Air Traffic Management in Europe During the Calendar Year 2014, 2015, 2016, Performance Review Commission.
Fricke, H., Schultz, M., 2008. Improving aircraft turn around reliability. In: Proceedings of 3rd International Conference for Research in Air Transportation, pp. 335–343.
Fricke, H., Schultz, M., 2009. Delay impacts onto turnaround performance. In: 8th USA/Europe Air Traffic Management Research and Development Seminar, Napa, CA.
Fuchte, J., 2014. Enhancement of Aircraft Cabin Design Guidelines with Special Consideration of Aircraft Turnaround and Short Range Operations. Ph.D. Thesis, Technische Universität Hamburg-Harburg, Hamburg, Germany, April 2014.
Gerdes, I., Temme, A., Schultz, M., 2016. Dynamic airspace sectorization using controller task load. In: 6th SESAR Innovation Days. Delft.
Gerdes, I., Temme, A., Schultz, M., 2018. Dynamic airspace sectorisation for flight-centric operations. Transport. Res. Part C: Emerg. Technol. 95, 460–480.

Gers, F.A., Schmidhuber, J., Cummins, F., 2000. Learning to forget: continual prediction with LSTM. Neural Comput. 12, 2451–2471.
Graves, A., Fernandez, S., Schmidhuber, J., 2007. Multi-dimensional recurrent neural networks. In: Proc. International Conf. on Artificial Neural Networks (ICANN-2007) 4668, pp. 865–873.
Grunewald, E., 2016. Incentive-based slot allocation for airports. Transp. Res. Procedia 14, 3761–3770.
Gwynne, S.M.V., Senarath Yapa, U., Codrington, L., Thomas, J.R., Jennings, S., Thompson, A.J.L., Grewal, A., 2018. Small-scale trials on passenger microbehaviours during aircraft boarding and deplaning procedures. J. Air Transp. Manage. 2018 (67), 115–133.
Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.
Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J., 2001. Gradient Flow in Recurrent Nets: the Difficulty of Learning Long-Term Dependencies. IEEE Press.
Ivanov, N., Netjasov, F., Jovanovic, R., Starita, S., Strauss, A., 2017. Air Traffic Flow Management slot allocation to minimize propagated delay and improve airport slot adherence. Transp. Res. Part A 95, 183–197.
Jaehn, F., Neumann, S., 2015. Airplane boarding. Eur. J. Oper. Res. 244 (2), 339–359.
Kafle, N., Zou, B., 2016. Modeling flight delay propagation: A new analytical-econometric approach. Transp. Res. Part B 93, 520–542.
Karlaftis, M.G., Vlahogianni, E.I., 2011. Statistical methods versus neural networks in transportation research: Differences, similarities and some insights. Transport. Res. Part C: Emerg. Technol. 19 (3), 387–399.
Kierzkowski, A., Kisiel, T., 2017. The human factor in the passenger boarding process at the airport. Procedia Eng. 187, 348–355.
Kingma, D.P., Ba, J., 2014. Adam: a method for stochastic optimization. In: Proceedings of the 3rd International Conference on Learning Representations (ICLR).
Li, Q., Mehta, A., Wise, A., 2007. Novel approaches to airplane boarding. UMAP J. 28 (3), 353–370.
Lv, Y., Duan, Y., Kang, W., Li, Z., Wang, F.Y., 2015. Traffic flow prediction with big data: a deep learning approach. IEEE Trans Intell. Transp. Syst. 16, 865–873.
Maa, X., Tao, Z., Wang, Y., Yu, H., Wang, Y., 2015. Long short-term memory neural network for traffic speed prediction using remote microwave sensor data. Transp. Res. Part C 54, 187–197.
Milne, R.J., Kelly, A.R., 2014. A new method for boarding passengers onto an airplane. J. Air Transport. Manage. 34, 93–100.
Milne, R.J., Salari, M., 2016. Optimization of assigning passengers to seats on airplanes based on their carry-on luggage. J. Air Trans. Manage. 54, 104–110.
Mirkovic, B., Vidosavljevi, A., Tosic, V., 2016. A tool to support resource allocation at small-to-medium seasonal airports. Journal of Air Transport Management 53, 54–64.
Mirza, M., 2008. Economic impact of airplane turn-times. AERO Quart. 4, 14–19.
Miura, A., Nishinari, K., 2017. A passenger distribution analysis model for the perceived time of airplane boarding/deboarding, utilizing an ex-Gaussian distribution. J. Air Transp. Manage. 59, 44–49.
Montlaur, A., Delgado, L., 2017. Flight and passenger delay assignment optimization strategies. Transp. Res. Part C 81, 99–117.
Mota, M.M., Boosten, G., De Bock, N., Jimenez, E., Pinho de Sousa, J., 2017. Simulation-based turnaround evaluation for Lelystad Airport. J. Air Trans. Manage. 64, 21–32.
Mueller, E.R., Chatterji, G.B., 2002. Analysis of aircraft arrival and departure delay. In: AIAA Aviation Technology, Integration, and Operations Conference, Los Angeles, CA.
Niklaß, M., Lührs, B., Grewe, V., Dahlmann, K., Luchkova, T., Linke, F., Gollnick, V., 2017. Potential to reduce the climate impact of aviation by climate restricted airspaces. Transp. Policy.
Nyquist, D.C., McFadden, K.L., 2008. A study of the airline boarding problem. J. Air Transp. Manage. 14, 197–204.
Okwir, S., Ulfvengren, P., Angelis, J., Ruiz, F., Guerrero, Y.M.N., 2017. Managing turnaround performance through Collaborative Decision Making. J. Air Trans. Manage. 58, 183–196.
Oreschko, B., Schultz, M., Fricke, H., 2011. Skill analysis of ground handling staff and delay impacts for turnaround modeling. Air Trans. Oper. 310–318.
Oreschko, B., Kunze, T., Schultz, M., Fricke, H., Kumar, V., Sherry, L., 2012. Turnaround prediction with stochastic process times and airport specific delay pattern. In: International Conference on Research in Airport Transportation (ICRAT), Berkeley.
Polson, N.G., Sokolov, V.O., 2017. Deep learning for short-term traffic flow prediction. Transp. Res. Part C 79, 1–17.
Qiang, S.-J., Jia, B., Xie, D.-F., Gao, Z.-Y., 2014. Reducing airplane boarding time by accounting for passengers' individual properties: a simulation based on cellular automaton. J. Air Trans. Manage. 40, 42–47.
Reitmann, S., Nachtigall, K., 2017. Applying Bidirectional Long Short-Term Memories (BLSTM) to performance data in air traffic management for system identification. In: Lintas, A., Rovetta, S., Verschure, P., Villa, A. (Eds.), Artificial Neural Networks and Machine Learning – ICANN 2017. ICANN 2017. Lecture Notes in Computer Science 10614, pp. 528–536.
Reitmann, S., Schultz, M. 2017. Stability analysis of recurrent neural networks for time series based system control in aviation. International Science & Progress Conference, Saint-Petersburg.
Reitmann, S., Schultz, M., 2018. Artificial data analysis in air traffic management. Council of European Aerospace Societies (CEAS) Aeronautical Journal (revision).
Reitmann, S., Schultz, M., 2018. Real-time Prediction of Aircraft Boarding. In: 37th AIAA/IEEE Digital Avionics Systems Conference (DASC), London.
Rosenow, J., Lindner, M., Fricke, H., 2017a. Impact of climate costs on airline network and trajectory optimization: a parametric study. CEAS Aeronaut. J. 8 (2), 371–384.
Rosenow, J., Fricke, H., Schultz, M., 2017b. Air traffic simulation with 4d multi-criteria optimized trajectories. Proc. of WSC 2017, 2589–2600.
Rosenow, J., Fricke, H., Luchkova, T., Schultz, M., 2018. Minimizing contrail formation by rerouting around dynamic ice-supersaturated regions. AAOAJ 2 (3), 105–111.
Rosenow, J., Schultz, M., 2018. Coupling of turnaround and trajectory optimization in an air traffic simulation. Proc. of WSC 2018 (in press).
SESAR, 2014. Concept of Operations Step 2 (Ed. 01.01.00). Single European Sky ATM Research Programme.
Schmidt, M., 2017. A review of aircraft turnaround operations and simulations. Prog. Aerosp. Sci. 92, 25–38.
Schmidt, M., Paul, A., Cole, M., Ploetner, K.O., 2016. Challenges for ground operations arising from aircraft concepts using alternative energy. J. Air Trans. Manage. 56, 107–117.
Schmidt, M., Heinemann, P., Hornung, M., 2017. Boarding and turnaround process assessment of single- and twin-aisle aircraft. In: 55th AIAA Aerospace Sciences Meeting, AIAA SciTech Forum, (AIAA 2017-1856).
Schmidt, M., Nguyen, P., Hornung, M., 2015. Novel aircraft ground operation concepts based on clustering of interfaces. SAE Technical Paper 2015-01-2401.
Schultz, M., 2010. Entwicklung eines individuenbasierten Modells zur Abbildung des Bewegungsverhaltens von Passagieren im Flughafenterminal. PhD thesis. TU Dresden, http://nbn-resolving.de/urn:nbn:de:bsz:14-qucosa-85592.
Schultz, M., 2014. Stochastic transition model for pedestrian dynamics. In: Weidmann, U., Kirsch, U., Schreckenberg, M. (Eds.), Pedestrian and Evacuation Dynamics 2012. Springer, pp. 971–986.
Schultz, M., 2017. Dynamic change of aircraft seat condition for fast boarding. J. Transport. Res. Part C: Emerg. Technol. 85, 131–147.
Schultz, M., 2018a. Field trial measurements to validate a stochastic aircraft boarding model. Aerospace 5 (1), 27.
Schultz, M., 2018b. Implementation and application of a stochastic aircraft boarding model. J. Trans Res. Part C: Emerg. Technol. 90, 334–349.
Schultz, M., 2018c. A metric for the real-time evaluation of the aircraft boarding progress. J. Transport. Res. Part C: Emerg. Technol. 86, 467–487.
Schultz, M., 2018d. Fast aircraft turnaround enabled by reliable passenger boarding. Aerospace 5 (1), 8.
Schultz, M., Schulz, C., Fricke, H., 2008. Efficiency of aircraft boarding procedures. In: Proceedings of the 3rd International Conference on Research in Air Transportation, pp. 371–377.
Schultz, M., Kretz, T., Fricke, H., 2010. Solving the direction field for discrete agent motion. ACRI 2010. Lecture Notes in Computer Science 6350, pp. 489–495.
Schultz, M., Fricke, H., 2011. Managing passenger handling at airport terminal. In: 9th USA/Europe Air Traffic Management Research and Development Seminar, Berlin.
Schultz, M., Kunze, T., Oreschko, B., Fricke, H., 2013a. Microscopic process modelling for efficient aircraft turnaround management. International Air Transport and Operations Symposium, Delft, The Netherlands.

Schultz, M., Kunze, T., Fricke, H., 2013b. Boarding on the critical path of the turnaround. 10th In: USA/Europe Air Traffic Management Research and Development Seminar, Chicago.
Soolaki, M., Mahdavi, I., Mahdavi-Amiri, N., Hassanzadeh, R., Aghajani, A., 2012. A new linear programming approach and genetic algorithm for solving airline boarding problem. Appl. Math. Model. 36, 4060–4072.
Standfuss, T., Gerdes, I., Temme, A., Schultz, M., 2018. Dynamic airspace optimisation. CEAS Aeronaut. J. 9 (3), 517–531.
Steffen, J.H., 2008. Optimal boarding method for airline passengers. J. Air Transport. Manage. 14, 146–150.
Steffen, J.H., Hotchkiss, J., 2012. Experimental test of airplane boarding methods. J. Air Trans. Manage. 18, 64–67.
Tielrooij, M., Borst, C., van Paassen, M.M., Mulder, M., 2015. Predicting arrival time uncertainty from actual flight information. In: 11th USA/Europe Air Traffic Management Research and Development Seminar, Lisbon, Portugal.
Verdonk Gallego, C.E., Gómez Comendador, V.F., Sáez Nieto, F.J., Orenga Imaz, G., Arnaldo Valdés, R.M., 2018. Analysis of air traffic control operational impact on aircraft vertical profiles supported by machine learning. Transport. Res. Part C: Emerg. Technol. 95 (2018), 883–903.
Wang, K., Ma, L., 2009. Reducing boarding time: synthesis of improved genetic algorithms. In: Yu, F., Zeng, J., Yue, G. (Eds.), Proceedings of the 2009 International Symposium on Web Information Systems and Applications, pp. 359–362.
Wang, Z., Liang, M., Delahaye, D., 2018. A hybrid machine learning model for short-term estimated time of arrival prediction in terminal manoeuvring area. Transport. Res. Part C: Emerg. Technol. 95, 280–294.
Yu, L., Wang, S., Keung Lai, K., 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. J. Air Transport. Manage. 58, 144–151.
Zhang, Z., He, Q., Gao, J., Ni, M., 2018. A deep learning approach for detecting traffic accidents from social media data. Transp. Res. Part C 86, 580–596.
Zhong, R.X., Luo, J.C., Cai, H.X., Sumalee, A., Yuan, F.F., Chow, A.H.F., 2017. Forecasting journey time distribution with consideration to abnormal traffic conditions. Transp. Res. Part C 85, 292–311.
Zhou, M., Qu, X., Li, X., 2017. A recurrent neural network based microscopic car following model to predict traffic oscillation. Transp. Res. Part C 84, 245–264.
Zhu, Y.B., Maoa, Y., Bai, S.Chen., 2017. A bi-level model for single-line rail timetable design with consideration of demand and capacity. Transp. Res. Part C 85, 211–233.