



Robust berth scheduling using machine learning for vessel arrival time prediction

Lorenz Kolley¹ · Nicolas Rückert¹ · Marvin Kastner² · Carlos Jahn² · Kathrin Fischer¹

Accepted: 1 August 2022 / Published online: 1 September 2022
© The Author(s) 2022

Abstract

In this work, the potentials of data-driven optimization for the well-known berth allocation problem are studied. The aim of robust berth scheduling is to derive conflict-free vessel assignments at the quay of a terminal, taking into account uncertainty regarding the actual vessel arrival times which may result from external influences as, e.g., cross wind and sea current. In order to achieve robustness, four different Machine Learning methods—from linear regression to an artificial neural network—are employed for vessel arrival time prediction in this work. The different Machine Learning methods are analysed and evaluated with respect to their forecast quality. The calculation and use of so-called dynamic time buffers (DTBs), which are derived from the different AIS-based forecasts and whose length depends on the estimated forecast reliability, in the berth scheduling model enhance the robustness of the resulting schedules considerably, as is shown in an extensive numerical study. Furthermore, the results show that also rather simple Machine Learning approaches are able to reach high forecast accuracy. The optimization model does not only lead to more robust solutions, but also to less actual waiting times for the vessels and hence to an enhanced service quality, as can be shown by studying the resulting schedules for real vessel data. Moreover, it turns out that the accuracy of the resulting berthing schedules, measured as the deviation of planned and actually realisable schedules, exceeds the accuracy of all forecasts which underlines the usefulness of the DTB approach.

Keywords Berth allocation problem · Machine learning · Uncertainty · (Arrival time) prediction · Robust optimization

✉ Lorenz Kolley
Lorenz.kolley@tuhh.de

¹ Institute for Operations Research and Information Systems, Hamburg University of Technology, Am Schwarzenberg-Campus 4, 21073 Hamburg, Germany

² Institute of Maritime Logistics, Hamburg University of Technology, Am Schwarzenberg-Campus 4, 21073 Hamburg, Germany

1 Introduction

A highly relevant issue in the planning of (maritime) supply chains and logistics is the uncertainty of deep sea vessel arrivals at ports, as its consequences affect all further stages of the supply chain (Dobrkovic et al. 2016). However, the port terminals are affected most: Berth allocations have to be planned in advance, i.e., while a vessel has not yet arrived and hence either a berth can still be occupied when a vessel arrives early (and the vessel cannot moor as planned) or a berth remains empty when a vessel is delayed. Both effects have negative economic consequences, for the terminal as well as for the vessel owners and, as mentioned above, also for the subsequent supply chain. Therefore, approaches for reducing uncertainty in this field are of huge importance.

As data and digitalization offer new opportunities for uncertainty reduction and mitigation, new approaches for terminal planning and management should be developed from a “data-driven perspective,” as Heilig et al. (2020) point out, in order to enrich the traditional “optimization perspective”. However, currently these authors still identify “a lack of data-driven approaches in the context of container terminals.” Moreover, they point out that data from external sources— as, e.g., Automated Information System (AIS) data—are “under-analysed”, i.e., that often they are not yet fully exploited; instead, the focus is often still on operational optimization, without sufficient consideration of patterns in existing data. According to Yang et al. (2019), Data Mining-for extraction of the relevant information-and vessel behaviour analysis-in order to find patterns of maritime traffic-are of high importance for future research in the field, as, e.g., causality analysis builds up on it. They point out that there is only little work in Operations Research (OR) yet which makes use of AIS data and that berth allocation might be one field which could strongly benefit from their exploitation in the future.

Therefore, this work is aimed at filling this gap with respect to a specific problem by developing and presenting a new machine-learning-based approach for forecasting vessel arrival times. The results are then used in a berth allocation approach, in order to improve the “traditional” berth allocation problem (BAP) optimization procedure by an appropriate use of existing data. The forecasting approach makes use of AIS data which are broadcast by vessels and, among other information, provide the current position of a vessel on a regular basis. Following Dobrkovic et al. (2016), data from 48 h before the actual vessel arrivals are used for forecasting, as this is the time frame within which vessel arrivals are usually announced to the respective seaport container terminal.

Through the use of real AIS data from the past on which the forecasts are based, but for which the actual vessel arrival times are already known as well, the approach taken in this work allows an evaluation of the forecast quality of the different forecasting methods. In the forecasting, methods from the field of machine learning (ML) are used, namely linear regression (LR), k -nearest neighbour (k NN), decision tree regressor (DTR) and artificial neural networks (ANN).

A berthing schedule is called (feasibility-)robust, when it remains stable at least for smaller deviations from the originally assumed vessel arrival times, i.e.,

when the schedule stays feasible (Scholl 2001). To derive a robust berth allocation based on the forecasts, the concepts of conflicts (Liu et al. 2017) and of dynamic time buffers (DTBs) are used (Kolley et al. 2021) and further refined. More specifically, Liu et al. (2017) define and measure robustness as a function of the service level that is achieved, where the service level in turn is defined as the number (or percentage) of vessels which-in the planned berth allocation-are not in conflict with other vessels. In this work, this customer-oriented concept is extended by the integration of time buffers which help to better prevent conflicts even if actual vessel arrival times deviate from scheduled berthing times. Hence, a *robust schedule* is defined as a schedule with as little conflicts as possible which remains valid under changing arrival times.

The purpose of the approach taken in this work is to reduce uncertainty and thus, increase robustness by means of using different forecasts-resulting from different ML methods as mentioned above-jointly in the berth scheduling. The forecasts are interpreted as possible future arrival times which are equally likely, hence uncertainty is transformed into risk. The DTBs are then constructed based on the resulting “forecast distributions”. The more the forecasts differ, i.e., the more spread there is in the distribution, the higher is the uncertainty with respect to the respective arrival time and hence, a larger buffer is needed for mitigating this uncertainty. On the contrary, with very similar forecasts, uncertainty can be assumed to be low (which is why all methods lead to similar results), there is little spread and hence, only a small buffer is necessary. The aim of the BAP approach then is to avoid conflicts, i.e., overlaps, not only for the planned berths, but also with respect to the time buffers, in order to enhance the schedule’s robustness.

For this purpose, the BAP model using DTBs from Kolley et al. (2021) is further modified and improved in this work. It is then applied to the forecast data. As the respective model builds up on the model of Liu et al. (2017), on their scenario-based approach and their robustness definition, the berth allocations which are derived using the model presented below are compared to the results from Liu et al.’s model in order to study the impact of the DTBs and the other changes made. The differences of the models are discussed in detail in Sect. 4.3.

The approach chosen in this work-the combination of AIS data exploration for forecasting, the subsequent use of an optimization approach which is based on the respective forecasts and the judgement of the quality of the results based on actual vessel arrivals-is, to the authors’ knowledge, unique, even if some publications use AIS data for forecasting. The contributions of this work hence are manifold:

- a) It is discussed and shown how AIS data can be used for arrival time forecasting and how the data sets can be cleaned and prepared beforehand. As Heilig et al. (2020) point out, such “methodological insights regarding data preparation (e.g., data cleansing, feature selection) [...] and model evaluation, are not discussed in great detail in literature”; hence, this fills an important gap.
- b) The different ML methods which are applied in this work are studied with respect to their forecast quality in order to judge which of them might be most appropriate for vessel arrival time prediction.

- c) It is shown how the forecast data can be used to derive more robust berth allocations by applying the concept of DTBs.
- d) Moreover, the BAP model from Kolley et al. (2021) is further refined such that the solutions are of more practical use and relevance. This is shown by comparing and benchmarking the resulting solutions with the model of Liu et al. (2017). The latter model is used as a benchmark as its robustness concept, which concentrates on the service level achieved in terms of vessels that can be served without conflict, is the concept which, with further refinements, is also used in this work.
- e) The use of real data moreover allows for studying the true service level, i.e., the resulting service level considering the real – not only the predicted – arrival times of the vessels. Hence, the evaluation of the BAP models has high practical (and not only theoretical) relevance.

In summary, the aim of this study is therefore to develop and examine a new procedure that generates robust berthing plans by predicting vessel arrival times by combining ML and OR methods. However, it has to be noted that increased robustness can lead to losses in efficiency, as these two objectives are in conflict: More robustness can be achieved by reserving more time for each vessel, but then the quay's capacity might not be fully exploited and there will be idle times, impeding efficiency.

The remainder of this paper is structured as follows: Literature on trajectory forecasting with ML in maritime logistics as well as on the BAP is reviewed in Sect. 2. Data pre-processing and the results of the ML algorithms are discussed in Sect. 3. In Sect. 4, the mathematical model for the robust continuous berth allocation problem with dynamic time buffers (ro-DTB-BAPc) is presented. The results of the numerical experiments are given and evaluated in Sect. 5. Finally, the paper is concluded by a critical discussion of the results, a summary and an outlook in Sect. 6.

2 Literature review

2.1 AIS Data and machine learning in maritime logistics

AIS was developed in the 1990's in order to improve maritime safety. Vessels regularly broadcast information via AIS, which can be split into the categories of static (e.g., vessel identity and size), dynamic (vessel's position, speed etc.) and voyage-related information (destination port and ETA), see Yang et al. (2019).

Yang et al. (2019) give an overview on maritime problems in which AIS data are relevant and they review different AIS data applications. In their review, Yang et al. state that AIS data were first mostly used in trajectory extraction and clustering for studies on navigation safety and the avoidance of vessel collisions, which they call "basic applications", but over time also "extended" and "advanced" applications in other areas of maritime research were developed. In their literature review, the authors find a strongly increasing number of publications on such AIS data applications in particular in the more recent past, i.e., since 2015.

More involved (extended) problem fields in which AIS data were successfully used are vessel behaviour analysis, e.g., with respect to fishing or to investigate navigation patterns or travel times and environmental evaluations, e.g., the analysis of emissions and, subsequently, the optimization of sailing speed in order to reduce such emissions.

AIS data are also used to analyse, in particular, activities at and near ports. For example, Wu et al. (2020) discuss quality issues of AIS data and in particular the problems these data can present with respect to identifying ports, i.e., how a “port event” and the fact if a vessel is actually at a port can be identified from AIS data. Feng et al. (2020) also use AIS data to derive trajectories in ports, while Chen et al. (2020) exploit AIS data to identify the services and analyse the characteristics of tugboat activities in ports. Franzkeit et al. (2020) investigate vessel waiting times at ports based on AIS data.

ML recently has been applied in Maritime Logistics in very different ways and to different planning problems. A structured review of the literature in this field is given by Dornemann et al. (2020); these authors also present an in-depth discussion of the combination of OR and ML methods. Moreover, Filom et al. (2022) present an extensive review of ML applications in ports. Therefore, just a few selected recent examples on work relating to berthing and the BAP are given below:

Li and He (2020, 2021) predict liner berthing times using deep learning. While Li and He (2020) present a more basic discussion of their approach, the data preprocessing and some preliminary results, their procedure is refined in the 2021 paper where it is shown that feature extraction has a huge impact on prediction results. The use of Deep Learning and an ANN for automatic berthing systems are suggested by Lee et al. (2020), who combine AIS data with actual ML. The authors use nine different supervised ML methods for “predicting the risk range of an unsafe berthing velocity when a ship approaches a port”, i.e., they study a very specific problem.

De Leon et al. (2017) employ ML for algorithm selection, i.e., for a meta-learning problem, to solve the Bulk Carrier BAP. By their approach, the best algorithm for the problem setting at hand can be determined, after the meta-algorithm has been trained on different problem settings and the relevant features having an influence on algorithm performance have been found. In contrast, Cheimanoff et al. (2021) use ML for parameter tuning of the hyper-parameters that are used in meta-heuristics for the Bulk Carrier BAP.

With respect to the topic of this work, it should be mentioned that De Leon et al. (2017) as well as Cheimanoff et al. (2021) and many other authors in the field use randomly generated vessel arrival times in their computational studies, i.e., no real data are used and no arrival time prediction is employed, even in the most recent publications. Consequently, no comparison with actual vessel arrival times can be made, as it is the case in this work.

2.2 Trajectory forecasting

Many studies in the field of maritime logistics estimate the time needed for given routes under the assumption that the vessels sail at a known and given speed (Grida

and Lee 2018). However, there are also studies using more involved approaches; many of them are based on the development of trajectories. Trajectories consist of a geospatial coordinate set and a time stamp. Zheng (2015) gives a broad overview on trajectory data mining in general and on the aspects of trajectory uncertainty, trajectory pattern mining, trajectory classification and trajectory outlier detection.

Dobrkovic et al. (2016) present a literature review on trajectory and arrival time prediction for deep sea vessels. In their work, they categorise the relevant publications into two groups, namely short-term predictions (up to 1 h) which are often made for collision avoidance and long-term predictions (more than 1 h) as they are required for the purposes of this work. It turns out that in the latter case, usually route extraction is applied, i.e., the vessels' "typical" sea lanes are determined and then each vessel under consideration is assigned a probability that the respective route is used. By this approach, anomalies (i.e., deviations from "normal" routes) can be detected. However, none of the approaches discussed in this review actually derives arrival time predictions.

Dobrkovic et al. also raise the issue of missing or erroneous data and they mention the importance of weather data, as due to different weather conditions, vessels often have to use different routes. However, Parolas (2016) finds in his study on AIS- and weather-data-based arrival time forecasting that the inclusion of weather data "was not of significant importance for estimating the time of vessel arrivals at the port."

In their subsequent work, Dobrkovic et al. (2018) focus on trajectory and arrival time prediction based on maritime pattern extraction. Their approach is based on sequential waypoint discovery, i.e., points through which many vessels travel, from which lanes / trajectories are derived, using a genetic, i.e., a "learning", algorithm.

The work of the above-mentioned authors is based on Pallotta et al. (2013) who use an unsupervised learning approach for detecting maritime traffic routes (trajectories) and travel patterns, for the determination of anomalies and for route prediction from AIS data. Also Kwun and Bae (2021) use AIS data to predict vessel travel paths and they also explicitly derive arrival time predictions. Their method uses a modified shortest path procedure, the A*-procedure, which is carried out on a grid structure. Based on the optimal path, the expected travel time and, hence, arrival time at a port can be calculated. Park et al. (2021) employ a similar approach.

Course pattern extraction from AIS data is also done by Fujino et al. (2018) who use a topic model which is usually applied in language processing and recognition. Their main purpose is to find anomalies and to detect off-course situations. Arguedas et al. (2018) use AIS data to develop maritime traffic networks, i.e., networks of shipping lanes, in order to enable the monitoring of vessel traffic, while Xiao et al. (2017) develop a maritime traffic forecasting methodology for short-term predictions by analysing past patterns of vessels' waterways and motion behaviour using AIS data.

2.3 Arrival time prediction without trajectories

Parolas (2016) is one of the few contributions which, without calculating trajectories, develops arrival time predictions. His work concentrates on the Asia-Rotterdam

route and uses two ML approaches for prediction, namely ANNs and support vector machines. Parolas states that “One of the most important findings [...] was that the AIS data alone are enough for making ETA predictions for the route and time-horizon examined.” Predictions could be significantly improved by ML, compared to the approaches which were used at Rotterdam port at the time.

Mestl and Dausendschön (2016) do not only predict ETAs, but also the port that a vessel is heading to, based on probabilities regarding vessels’ routes which they derive from AIS data-based knowledge about the routes other (similar) vessels took before and the ports they headed for from a certain position. The information is gained from AIS data which, according to the authors, often do not contain the correct entry of the next destination port.

Grida and Lee (2018) estimate sailing and berthing times based on AIS data. First, the AIS data are pre-processed, then the estimation is done by multi-linear regression. Variables used in the linear regression for travel time estimation are, among others, distance to port and size of vessel. They achieve very good results in terms of R^2 values reached and hence, their approach seems to be particularly promising. Therefore, linear regression is one of the approaches that will be also used in this work.

Fancello et al. (2011) use ANNs to forecast vessel arrival times and subsequently optimize the allocation of workers to shifts at the respective port based on the predictions. As they point out, it was shown by Zhang et al. (1998) that ANNs perform better in forecasting than traditional classical methods when time series are rather irregular; hence they are suited well for the situation at hand. By testing different network configurations, the authors are able to reduce the uncertainty interval of arrival times considerably (from about 8 to about 6 h); this is useful for the subsequent shift planning at the terminal. However, a range of about 6 h within which a vessel is expected to arrive still makes the berth allocation planning rather difficult. Hence, the predictions need to be further improved and uncertainty intervals need to be further reduced for the purpose of this work.

Pani et al. (2014) use Data Mining and, in particular, a classification and regression tree model in order to forecast vessel delays for the harbour of Cagliari within a short time horizon; these authors achieve promising results by their vessel classification method with a mean error of about 1.5 h for delay prediction.

Pani et al. (2015) employ different ML approaches for predicting if vessels will arrive early or will be delayed. As Pani et al. (2014), they do not base their predictions on AIS position data, but on features which classify the vessels like, e.g., their length; moreover, they also include weather data. According to these authors, both these aspects turn out to have a relevant influence on delays.

It can be concluded from this literature review that the approaches taken so far in the literature are in many cases based on trajectory or lane prediction from which then an arrival time can be deduced, with Parolas (2016) and Grida and Lee (2018) being notable exceptions who base their predictions on AIS data. Building up on their work, the approach suggested in this paper is the application of different ML methods to AIS data in order to directly derive arrival time predictions from them.

2.4 Berth allocation planning under uncertainty

There is a vast body of literature on the BAP and its different variants. Literature reviews, a classification scheme and many details on the BAP can be found in, e.g., Bierwirth and Meisel (2010, 2015) and Carlo et al. (2015).

In this work, the dynamic BAP with a continuous quay and deterministic (fixed) handling times, but with uncertain vessel arrival times is studied. Thus, the following review puts the focus on the BAP under uncertainty and more specifically on proactive approaches for these problems and the use of time buffers. A structured overview of relevant literature on the BAP under uncertainty can be found in Liu et al. (2020).

A concept which is suggested by many authors for handling uncertainty in the BAP is the scenario-based approach. It is chosen, e.g., by Hendriks et al. (2010), Xiang et al. (2017) and Liu et al. (2017). All these publications take a proactive approach. While the latter two take uncertain travel times and uncertainty in handling times into account, Hendriks et al. (2010), who study a strategic problem, only consider uncertainty in handling times.

A two-stage proactive–reactive solution approach is taken by Zhen et al. (2011) and also by Liu et al. (2020). In addition to a proactive and scenario-based first-stage, they consider possible disruptions and recovery operations on the reactive second model stage. In particular, Liu et al. (2020) point out that usually the probabilities needed in a scenario-based approach are not known and hard to derive; hence, a robust approach for handling uncertainty is preferable.

Such a robust approach can be developed using time buffers. The concept of time buffers for mitigating uncertainty and in order to achieve robust BAP schedules was first suggested by Xu et al. (2012), who consider uncertain arrival and handling times. (The use of buffers is well known from other areas, as, e.g., flight scheduling (Baumgarten et al. 2014) or production management (Leus and Herroelen 2007), where the buffers help in deriving robust schedules as well.) In contrast, Wu and Miao (2020) do not use time buffers but space buffers, i.e., they do not determine a fixed position for each vessel but add some “slack” to the planned berths and hence aim for robustness by enhancing flexibility in terms of space. While they can show that this leads to a reduction in expected waiting times and, therefore, expected costs, this may impede efficiency when too much space and capacity is reserved for each vessel.

The Berth Allocation and Quay Crane Assignment Problem (BQCAP) with uncertain arrival times is studied, e.g., by Wang and Guo (2018). They proactively consider uncertain arrival times and aim at a robust schedule for berths and quay cranes. Rodriguez-Molins et al. (2014a, 2014b) consider uncertain handling times and use variable time buffers in their approach for the BQCAP, while Li et al. (2019) suggest a proactive scenario-based approach for this problem. Both uncertain arrival and handling times in the BQCAP are studied by Zhang et al. (2014) who proactively assign time buffers to vessels; however, the time buffer for all vessels is identical in their approach.

It can be concluded that proactive approaches are useful to handle uncertainties in the dynamic BAP and time buffers can be employed to proactively increase

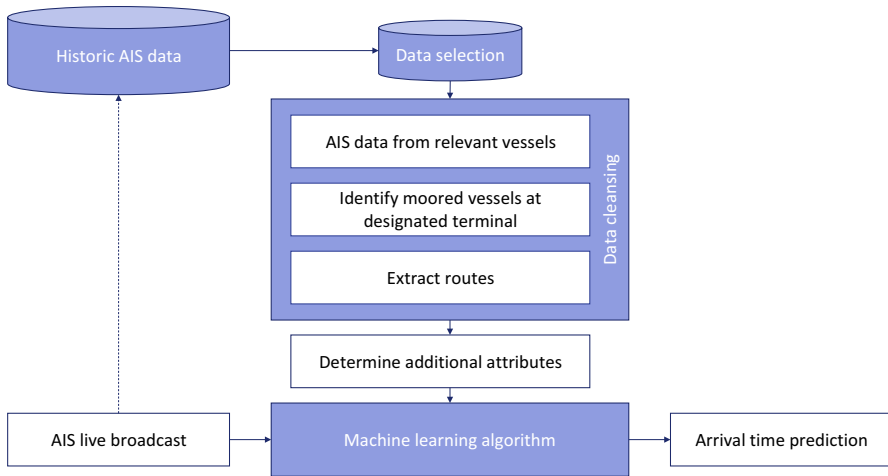


Fig. 1 Process of data pre-processing

a schedule's robustness. These features will be exploited in the approach taken in this work which is aimed at allocating individual time buffers to the different vessels, depending on the degree of uncertainty of their arrival time. This approach is described in Sect. 4, while Sect. 3 is dedicated to data and forecasting.

3 Data pre-processing

3.1 Data selection and preparation

The AIS data which is used in this work is provided by the US National Oceanic and Atmospheric Administration (NOAA) and already reduced to one AIS message per minute. The chronological sequence of the AIS messages is given by the current time stamp (BaseDateTime). Each AIS message contains several pieces of information: The Maritime Mobile Service Identity (MMSI) is a unique identification number of nine digits where the first three digits represent the Maritime Identification Digits (MID) (see Article 19.108A §41 in the Radio Regulations¹) of the administration (e.g., a country's government) at which the corresponding vessel is registered. For the purpose of berth assignment, the MID is limited to the range from 201 to 775 (cf. the Table of Maritime Identification Digits²) to exclude groups of vessels broadcasting together or other devices, e.g., coast stations. The subsequent six digits identify the specific vessel registered to the respective administration.

¹ International Telecommunication Union (2020): Radio Regulations.
URL: <https://www.itu.int/pub/R-REG-RR-2020>

² International Telecommunication Union (2017): Table of Maritime Identification Digits.
URL: <https://www.itu.int/gladapp/Allocation/MIDs>

Table 1 Value ranges and error codes of the relevant attributes

Attribute	Allowed ranges	Error code
MMSI	201,000,000–775,000,000	None
LAT	N 0° to N 50°	None
LON	E–140° to E–110°	None
SOG	0–102 knots	102.2 knots
COG	0°–359°	360°
Heading	0°–359°	511°
Vessel type	70–79	None
Status	0, 1 and 5	15

Each AIS message includes the current position in the form of latitudinal and longitudinal coordinates (LAT and LON). Additionally, the current speed over ground (SOG) is broadcasted. The course over ground (COG) is the direction in which the vessel moves whereas the heading is the direction in which the bow points (these may be different). With the attributes VesselType and Cargo, the precise type of the vessel and (if available) the kind of freight is specified. The status attribute indicates the current operational status of the vessel, i.e., whether it is, e.g., underway, undertakes fishing operation, or is moored at the quay. A vessel's dimensions are specified in terms of its length, width and draft.³

The overall process of deriving forecasts for vessel arrival times from past data is shown in Fig. 1. As Heilig et al. (2020) point out, the steps of selecting, pre-processing and transforming the relevant data-i.e., the data cleansing-are important before patterns in the data are to be extracted. In terms of data selection, it is recommendable to consider only recent entries from the available historic AIS data, as the overall world fleet's characteristics may change over time, e.g., due to a larger proportion of vessels with higher transport capacities. Hence, AIS data from 2018 on are used in this study. Moreover, only those data are relevant which concern trips ending at the respective port terminal.

The data cleansing is critical to the accuracy of the resulting forecasts, because the ML algorithm can only learn from the information included. Therefore, providing good data quality is the first step whenever ML is to be applied and it is particularly important for ANN as pointed out by Fancello et al. (2011).

Therefore, the data cleansing for this application is carried out as follows: Firstly, in order to identify the relevant AIS data, the data is reduced to a suitable geographical area, the relevant types of vessel and cargo are set and empty or false entries are removed. In particular, AIS messages that do not include the values for all relevant attributes are deleted. The allowed ranges and error codes for the considered attributes are presented in Table 1. The error codes mark incomplete or missing entries by broadcasting a default value; in this case, the data is to be excluded.

³ Navigation Center (2021): Class A AIS Position Report.
URL: <https://www.navcen.uscg.gov/?pageName=AISMessagesA>

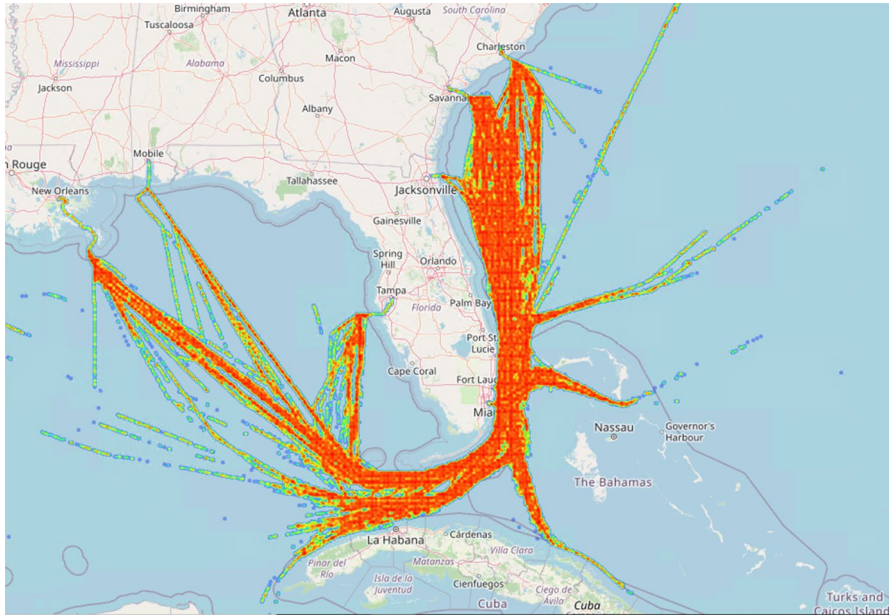


Fig. 2 Heatmap of port calls to Miami (Map data by © OpenStreetMap, under ODbL.)

The classification of the vessel type in the AIS data is not strict enough to clearly identify container vessels. The vessel types 70 to 79 describe general cargo vessels including container vessels, but are not limited to those.⁴ Therefore, without additional information, container vessels can only be identified based on the fact that a certain vessel moors at a container terminal's quay. Thus, the second task in the data cleansing is to extract those vessels that have moored at the designated container terminal already in the past.

The terminal's quay area is represented by a polygon. All AIS messages broadcasted with a current position inside the quay area and moreover with the status "5", corresponding to a vessel moored at the quay, are considered to be vessels arriving at the designated container terminal, i.e., a "port event". The first AIS message of each trajectory from inside the quay area ending at the designated container terminal is used to determine the arrival time of that vessel at the port. All other AIS messages also sending the status moored ("5"), but not located at the quay area, are deleted.

In order to enhance the prediction accuracy, AIS data describing the manoeuvring at the port, e.g., tug boat rides and mooring activities, within a distance of 6 nautical miles are excluded and the data is reduced to a maximum of 48 h of journey to the port.

⁴ Digital Coast (2018): AIS Vessel Type and Group Code.

URL: <https://coast.noaa.gov/data/marinecadastre/ais/VesselTypeCodes2018.pdf>

Knowing the actual arrival times of all calling vessels from the past data, the actual remaining travel time can be calculated for each AIS message from the vessels' port approach. To do so, the AIS messages of a specific vessel are brought into reverse chronological order and subsequently changes in the status attribute are analysed. As only instances from the quay area of the designated terminal can hold the status "moored", whenever the status changes from "moored" ("5") to "under way using engine" ("0"), this marks the end of the vessels approach to the port, i.e., the vessel's arrival at the terminal. From there, the AIS messages in a trajectory can be followed backwards until a disruption of more than two hours occurs in the data. As such disruptions signify the entrance into the area which is covered by shore AIS stations, a longer period of missing AIS messages or time that this vessel spent in other ports or at anchorage, then the trajectory ends. For arrival time prediction, the respective last (i.e., earliest) AIS message marks the beginning of the terminal approach. All other AIS messages not related to the vessel's approach are deleted. In addition to the remaining Euclidian distance to the designated terminal, also the attribute "drift", describing the discrepancy between COG and heading (Kolley et al. 2021), is derived from the AIS messages.

To illustrate the process, the relevant AIS messages for the Port of Miami from 2018 to 2020 are plotted on a map in Fig. 2.⁵ Obviously, vessels approach the Port of Miami from many origins. In the east and south of Florida, vessels operate in a rather wide area and are able to overtake each other. Thus, each vessel can operate at individual speed without considering other vessels, i.e., congestions are avoided. The trajectories show that the majority of the port approaches are from the north and east, while trajectories from the west are narrow and light in colour due to less approaches and therefore lower density of AIS messages. The more complex learning algorithms might exploit such properties, based on the vessels' coordinates.

Furthermore, approaches at other ports' terminals are visible that could not be identified and hence were not eliminated in the data cleansing. This is due to the low reliability of the AIS data (especially in the dynamic attributes) when the AIS status of these vessels was not set to "moored" ("5") at the respective ports. In Fig. 2, this effect can be seen at many different ports on the Atlantic coast and the Gulf of Mexico where the sequence of AIS messages begins at the sea, reaches a port, leaves the port again and eventually approaches the designated terminal. In order to rectify this effect, the other ports could be represented through polygons and AIS messages send from inside those polygons then could be deleted.

In this work, no additional data, e.g., weather data, is considered. However, external factors, i.e., sea current and crosswind, can influence the vessels speed and therefore the remaining travel time to the designated terminal (Pani et al. 2014). This effect is taken into account through the additional attribute "drift".

The processed data (Kolley et al. 2022) is split into a training dataset from January 1st 2018 until December 31st 2019. Hence, the training data consists of two

⁵ The raw data is available on: <https://marinecadastre.gov/ais/>.

A refined dataset is available on the TU Hamburg Open Research (TORE) repository under the following <https://doi.org/10.15480/336.4471>.

complete years, each including all four seasons. Therefore, the ML algorithms can learn the differing weather conditions throughout the years and the forecasts are not restricted to specific seasons. The remaining data from 2020 is again split into a validation dataset including January to June 2020 and a test data set from July to December 2020.

The ML models can be used to predict possible arrival times for these and other (unknown) AIS test data. They also could be used on a live feed of broadcasted AIS messages. When the currently approaching vessels arrive at the designated terminal, their AIS data can feed into the historic AIS database, such that it is used in future predictions (see Fig. 1).

3.2 Selection of ML methods for forecasting remaining travel time

When creating a berthing schedule, the time of arrival needs to be estimated for each vessel that will arrive during the planning horizon. This can be realized by using the current time as a base point in time and then treating the remaining travel time to the designated port as a regression problem (Jahn and Scheidweiler 2018). Thus, any regression method that can estimate the remaining travel time based on the available AIS data, is permissible. Over the last decades, a wide range of statistical models and ML algorithms have emerged that can be applied to the regression problem at hand. In the following, it is explained how the methods used in this work were selected.

As a baseline method, an LR model is chosen (see, e.g., Frochte 2019; Géron 2018). When predicting the remaining travel time, the distance of the vessel to the port is a major influencing factor. Further vessel-related and journey-related attributes from the AIS data can be additionally exploited to modify this estimation. When a vessel approaches a container terminal, its feature vector can be inserted into the fitted equation to obtain the prediction.

The k NN algorithm is used as the second method. This learning algorithm has been previously described by Langley and Simon (1995) and it is suitable for noisy real-life data, as it only considers those data which are close to a certain feature vector. It has been applied to AIS data before in the context of emission monitoring by Virjonen et al. (2018) in order to predict the future location of vessels in the Gulf of Finland. Compared to LR, the k NN algorithm takes a completely different approach. Instead of creating a model of the world (by, e.g., assuming a linear relationship between variables), this instance-based approach keeps historic data in memory. To predict the remaining travel time of a vessel, similar entries of past port approaches in the database are scanned. The similarity of two neighbours (i.e., vectors of attributes) is measured by the Euclidean distance and only the k most similar entries are taken into further consideration. The predicted remaining travel time is computed by averaging the remaining travel times of the k considered neighbours. While averaging, each of the neighbours can either be weighted uniformly or by the inverse of their distance to the point for which the prediction is run.

As a more sophisticated ML method, a DTR is chosen. Decision trees generally decide in each node with respect to which attribute the training data is to be divided.

This decision is made according to its features' characteristics in order to achieve the highest information gain regarding the estimation of the respective label, e.g., the remaining travel time. The leaves consist of data for which a further split will not generate any additional information for the prediction of the respective label (Bose and Mahapatra 2001; Frochte 2019; Yu et al. 2018). The DTR is a specialised type of the decision tree especially suitable for regression tasks. Using an LR model, the relation between features and labels is represented for the data in each leaf separately (Frochte 2019). The suitability of DTR for short-term arrival time prediction is shown by Pani et al. (2014).

In the last decades, major progress has been made in the research regarding ANNs (e.g., Géron 2018; Belousov et al. 2021). In this work, feed-forward ANNs are used (see e.g., Géron 2018). Such a ANN consists of one input layer, a chosen number of hidden layers and one output layer while each of the layers consists of a certain number of nodes. The number of neurons for the input layer is determined by the number of considered attributes, the number of hidden layers and the number of neurons for each of the hidden layers are chosen by the modeller and for regression problems exactly one neuron exists in the output layer. Each hidden layer is fully connected with its preceding and successive layer and each neuron is described by the formula $f(x) = a(bx_1 + w_2x_2 + \dots + w_nx_n)$, for which n refers to the number of neurons in the preceding layer, b and w_i are parameters that are fitted to the training data and a denotes the activation function.

The activation function is also chosen by the modeller prior to fitting the model to the data. In this work, all activation functions supported by *scikit-learn*⁶ are considered: the identity function $a(x) = x$, the rectified linear unit (ReLU) with $a(x) = \max(0, x)$, the hyperbolic tangent function $a(x) = \tanh(x)$ and the logistic function $a(x) = (1 + \exp(-x))^{-1}$. For the trivial case where no hidden layer exists and the identity function is selected as the activation function, the ANN corresponds to the LR. By adding hidden layers and choosing a slightly more complex activation function, such as *ReLU*, non-linear relationships between the different variables can be modelled as well (Schmidt-Hieber 2020). Furthermore, the two alternative solvers *adam* (Kingma and Ba 2015) and *lbfgs* (Liu and Nocedal 1989) are employed for tuning the weights of the neurons during the learning phase.

ANNs have been previously applied to the remaining travel time estimation problem with good results (e.g., Fancello et al. 2011; Jahn and Scheidweiler 2018). Fancello et al. (2011) further state that determining the hyper-parameters (e.g., the number of hidden layers, number of neurons for each hidden layer and the activation function) was the greatest challenge and that a trial-and-error procedure was followed in their work. The respective issue is tackled in this work by adding a second iterative layer around the ML phase to systematically optimize the hyper-parameters.

⁶ *scikit-learn* (2021): Machine Learning in Python. URL: <https://scikit-learn.org/>

Table 2 Considered value ranges for hyper-parameters and input data variations

Learning algorithm	Hyper-parameters	Scale	Permissible values
LR	Feature set	–	{Unscaled_reduced, unscaled_full, scaled_reduced, scaled_full}
	Feature set	–	{Unscaled_reduced, unscaled_full, scaled_reduced, scaled_full}
KNN	Number of neighbours	Log	[10, 500] \subset \mathbb{N}
	Weights	–	{Uniform, distance_based}
DTR	Feature set	–	{Unscaled_reduced, unscaled_full, scaled_reduced, scaled_full}
	Maximum depth	Linear	[10, 500] \subset \mathbb{N}
	Maximum number of features	Linear	[1, n_{features}] \subset \mathbb{N} with n_{features} being the number of features in the selected dataset
ANN	Minimum number of samples at a leaf	Log	[10, 5000] \subset \mathbb{N}
	Minimum number of samples at a split	Log	[10, 5000] \subset \mathbb{N}
	Feature set	–	{Unscaled_reduced, unscaled_full, scaled_reduced, scaled_full}
	Number of neurons in first hidden layer	Linear	[1100] \subset \mathbb{N}
	Number of neurons in second hidden layer	Linear	[1100] \subset \mathbb{N}
	Activation function	–	{Identity, logistic, tanh, ReLU}
	Solver	–	{Adam, lbfgs}
Learning rate	Log	[10^{-5} , 10^{-1}] \subset \mathbb{R}	

Table 3 Hyper-parameters leading to the highest R^2 values for the validation set, as identified by the TPE heuristic

Learning algorithm	Hyper-parameter	Selected value	R^2 validation set
LR	Feature set	Unscaled_reduced	0.61
kNN	Feature set	Unscaled_reduced	0.7
	Number of neighbours	500	
DTR	Weights	Distance_based	0.67
	Feature set	Unscaled_reduced	
	Maximum depth	15	
	Maximum number of features	3	
	Minimum number of samples at a leaf	2221	
ANN	Minimum number of samples at a split	136	0.71
	Feature set	Unscaled_reduced	
	Number of neurons in first hidden layer	22	
	Number of neurons in second hidden layer	63	
	Activation function	tanh	
	Solver	adam	
	Learning rate	2.43×10^{-4}	

3.3 Hyper-parameter optimization for ML methods

Generally, there are many design decisions to be made when employing ML methods. This also applies to the problem of forecasting remaining travel times. Except for LR, all learning algorithms that are used in this work are configurable, i.e., they use hyper-parameters which are chosen before training and which influence the resulting forecast quality. The selection of configuration options cannot be re-adjusted during the learning phase, as this would undermine the procedure.

As pointed out by different authors, the selection of appropriate hyper-parameters is a non-trivial problem (Bergstra et al. 2011; Bergstra and Bengio 2012; Eggensperger et al. 2013). Fancello et al. (2011) highlight the importance of variable normalisation, the choice of an appropriate learning algorithm and of its related hyper-parameters when forecasting the arrival times of vessels. In addition, they point out that using more attributes does not automatically lead to better results, a fact also known as the feature subset selection problem (Shukla et al. 2019; Takano and Miyashiro 2020).

The feature subset selection process and the choice of variable normalization are carried out as follows. Instead of covering all possible options, four feature sets have been defined which are named ‘*unscaled_reduced*’, ‘*unscaled_full*’, ‘*scaled_reduced*’ and ‘*scaled_full*’. Each reduced feature set contains the attributes LAT, LON, SOG, i.e., position, speed and the distance to the port. Each full feature set additionally includes the heading, drift, vessel length and its width. An unscaled feature set contains the respective information in its original units, i.e., time spans are reported in minutes, distances in nautical miles and coordinates in degrees.

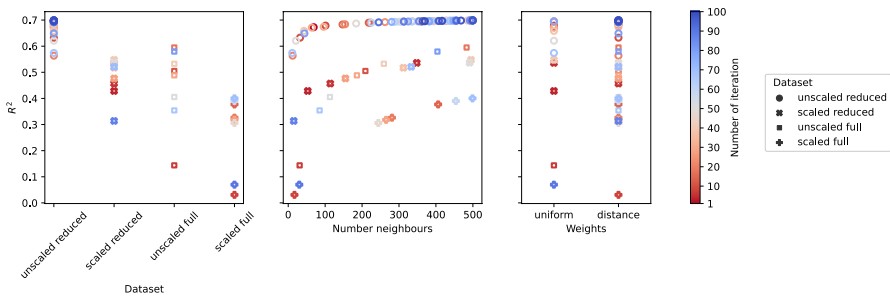


Fig. 3 Hyper-parameter selection process for k NN

In a scaled set, each attribute is scaled with a standard scaler. A vector of the true remaining travel times in unscaled format is maintained separately.

The four different options for the features to be used as well as the hyper-parameters that are varied are shown in Table 2. For the sake of simplicity, both are jointly referred to as “hyper-parameters” here. The number of possible combinations of hyper-parameters in Table 2 exceeds what can be reasonably evaluated: Even if the learning rate is neglected, the number of possible combinations exceeds $2.9e^{11}$ for the ANN alone. Thus, a reasonable subset of experiments must be selected. Bergstra and Bengio (2012) argue that this should not be done manually because even a random sampler outperforms researchers who use their intuition. However, the tree-structured Parzen estimator (TPE) has been shown to outperform a random sampling process (Bergstra et al. 2011; Eggenesperger et al. 2013). This approach has first been presented by Bergstra et al. (2011) and is employed in this work. It uses Bayesian statistics to update distributions and the expected improvement to identify promising new hyper-parameters for each iteration. Each hyper-parameter is modelled independently of each other; thus, all hyper-parameters are varied concurrently. To further steer the exploration process, some variables are first log-transformed. Such a transformation is indicated in the column “scale” in Table 2. For each hyper-parameter, initially a uniform distribution is assumed and a parameter realization is drawn randomly from this distribution.

For each learning algorithm, its *scikit-learn* implementation for Python is employed, using the default values of that library unless indicated otherwise. In the hyper-parameter optimization, the TPE implementation of the *optuna* library for Python is used (Akiba et al. 2019). Each ML method is given 100 iterations to maximize the R^2 value on the validation set; this turned out to be sufficient, as a pre-study with 1000 iterations showed no improvement at all for k NN and only minor improvements at the third decimal place for DTR and ANN. For each ML method, the hyper-parameters that lead to the highest R^2 value are then used for creating the vessel arrival time forecasts for the validation set.

The results of the hyper-parameter optimization process are presented in Table 3. In all cases, the best observed value for the feature set was ‘*unscaled_reduced*’. For ANNs, this contradicts the previous findings of Fancello et al. (2011) as well as the

Table 4 Error distribution and regression metrics of the ML algorithms for all test data

n = 96.217	Sd (min)	Median (min)	MAPE (%)	MAE (min)	RMSE (min)	R^2
LR	193.5	96.5	40.9	154.6	204.6	0.69
kNN	180.6	71.9	29.0	131.4	186.9	0.74
DTR	190.6	64.8	30.4	139.5	197.2	0.71
ANN	175.3	76.5	29.7	132.0	182.8	0.75

The best value for each column is set in bold

general guidelines of *scikit-learn*⁷ regarding scaling, while the result with respect to the feature set—the reduced set being preferable—is in accordance with their results and results regarding feature selection from other areas, e.g., for predictions in the field of renewable energy applications, esp. wind speed prediction (Salcedo-Sanz et al. 2018), while the situation is somewhat different in disease prediction where only “simple” diseases can be predicted using a small number of features (Chen et al. 2017).

Furthermore, *k*NN outperforms DTR and performs very close to ANN on the validation set, indicating that *k*NN as an instance-based learner is a suitable approach for estimating the remaining travel time.

It should be noted, however, that in some cases different constellations of hyper-parameters led to the same results, i.e. the “best” combination as presented in Table 3 is not unique. In particular, often the unscaled and scaled variant performed equally well, e.g., for LR. Moreover, the resulting hyper-parameters for ANN depend not only on the data used, but also, e.g., on the order in which the data are “fed” to the neural network. Hence, care must be taken with respect to generalizing the results.

For *k*NN, the hyper-parameter search process is discussed here in more detail to illustrate the procedure. Note that, however, the respective evaluations were carried out for all four methods.

In Fig. 3, for each hyper-parameter value its corresponding R^2 values are plotted as a slice plot. This visualization neglects the interactions of the hyper-parameters, but shows more clearly the influence of the individual parameter choices. Each marker indicates the selection of a value for the respective hyper-parameter. The colour bar indicates at which stage during the heuristic hyper-parameter optimization process the respective value was evaluated. For the first iteration (in dark red), the values were sampled randomly. As the colour turns into blue, more and more already probed hyper-parameter values are incorporated in the probabilistic model used by TPE, thus steering the process towards more promising hyper-parameter settings.

⁷ *scikit-learn* (2021): Neural network models.

URL: https://scikit-learn.org/stable/modules/neural_networks_supervised.html#tips-on-practical-use

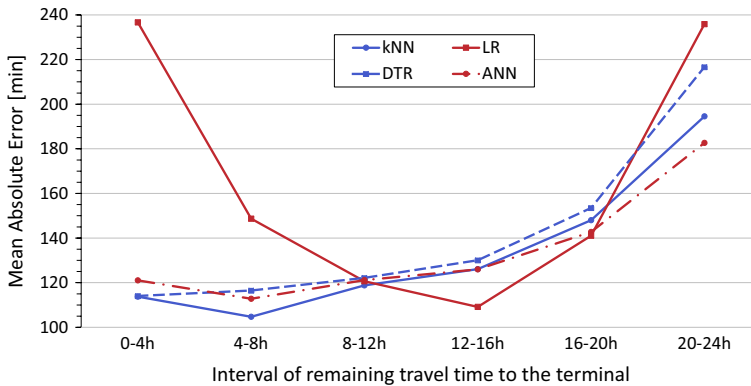


Fig. 4 MAE depending on remaining travel time

For the feature set selection, a clear preference for the unscaled reduced variant can be observed. The wide range between the lowest and highest R^2 values in the left-most dataset subplot indicates that (at least) one additional influencing factor is present beside the feature set selection. When the number of neighbours for each of the 100 iterations is plotted against the R^2 value in the second subplot, a curve-like shape is observed for each case, showing the relevance of this factor. In the value range under consideration, more neighbours result in better (or at least not worse) predictions. This indicates that the data is noisy, as more data points have to be averaged to achieve higher R^2 values. For the hyper-parameter ‘weight’, the last subplot is less obvious to interpret, but there seems to be a slight advantage of distance-based weighting.

In summary, the plots highlight the importance of proper feature selection and feature pre-processing as well as hyper-parameter optimization.

3.4 Results of the different ML methods

After the settings for the hyper-parameters have been determined using the validation set, next the performance of these hyper-parameter settings on the test data set is evaluated. In this work, only port approaches within the next 24 h, i.e., 1440 min, are considered. For all predictions on the test data, the error, i.e., the difference between the actual and the forecasted remaining travel time, is further examined. As a summary of the error distributions, the metrics Mean Absolute Percentage Error (MAPE), Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) are reported and the R^2 values are given in Table 4.

LR shows the worst performance over all four metrics, followed by DTR. The two approaches k NN and ANN perform best and are head-to-head with very small differences in their error metrics.

To analyse how prediction accuracy of the learned models depends on the vessels’ remaining travel time to the designated terminal at the port, in Fig. 4 the MAE

is shown for time intervals of 4 h for each ML approach. The methods k NN, DTR and ANN perform similarly in the considered time horizon of 24 h. Thus, the analysis of the three approaches' accuracy for different time intervals confirms the overall results in Table 4. Moreover, k NN and ANN show a constantly increasing MAE for increasing travel times from 4 to 8 h on. DTR shows the same behaviour already from the beginning. This is in line with the expectation that prediction accuracy increases with decreasing remaining travel time to the designated terminal. When a vessel is close to the terminal, many different factors influence the time it takes to actually get to the final berthing position, therefore the relative deviations in the data are bigger and forecast quality deteriorates again, leading to worse values for the 0–4 h interval.

However, as can be seen in Fig. 4, the LR method deviates from the behaviour of the other approaches. While for 0–4 h, 4–8 h and 20–24 h the MAE is much higher than for all other approaches, between 8 and 20 h remaining travel time to the terminal LR performs unexpectedly well. Obviously, in this range of travel times, the arrival time can be forecast best as a linear function of the vessel's position, distance to the terminal and speed, whereas this is not possible when a vessel is rather close to or still quite far from the terminal.

4 Robust berth scheduling

4.1 Dynamic time buffers

The model presented below considers dynamic vessel arrivals at a continuous quay with fixed handling times, with the goal to assign each incoming vessel to a specific berthing time and position and to achieve high service quality in order to satisfy the terminal's customers. The continuous quay layout allows the mooring of vessels at any position, but the quay's capacity is limited by the quay length and the considered planning horizon. If there is not enough quay space available at a specific time for a vessel to be moored, it cannot be assigned without conflicting with other vessels, or it needs to be postponed. The proportion of vessels that can be served without conflict is regarded as the achieved service level (following Liu et al. 2017) which is to be maximized. Moreover, further service quality measures are considered, e.g., the vessels' waiting times for their berths.

A vessel's arrival time is uncertain until the vessel actually arrives. However, the berth scheduling has to be conducted beforehand, when only estimations of arrival times can be considered. Therefore, the stability of the schedule is enhanced by adding time buffers between any two vessels, compensating for possible vessel delays. Consequently, the quay remains idle for that time, as the buffer blocks additional capacity for the related vessel. The more idle time is assigned, the higher is the robustness of the schedule, but the lower the quay's efficiency is expected to be. Among others, Xu et al. (2012) assign time buffers of the same length to all vessels, but the uncertainty in the arrival times is not necessarily identical. Hence, in their

approach a long idle time may also be assigned when uncertainty for a specific vessel in fact is low.

Therefore, aiming for a robust and efficient schedule, in this work this issue is addressed by defining individual time buffers which reflect the uncertainty in the arrival time of a specific vessel. Thus, less unnecessary idle time is assigned and efficiency can be maintained.

To predict possible arrival times for future incoming vessels, the ML algorithms presented in Sect. 3.2 are applied to derive forecasts based on knowledge from past AIS data. Due to the lack of certain information about the actual arrival time, the overall accuracy of the forecasts is unknown beforehand, i.e., when the scheduling is conducted. Thus, each forecast is interpreted as one possible future scenario and the scenarios are assumed to be equally likely.

Obviously, the ML algorithms can only learn information that is included in the training data. Therefore, the resulting prediction is expected to be more accurate for instances similar to the training data, i.e., in cases where the uncertainty is low; then, only a short time buffer is necessary. However, for new instances that behave differently, i.e., when uncertainty is high (e.g., when a vessel has to avoid bad weather and hence takes a detour), the forecast cannot be that precise. To ensure that the schedule is stable (and hence robust), in this case a longer time buffer is recommendable. The concept of DTBs (Kolley et al. 2021) for conflict avoidance in berth scheduling is based on these ideas. A DTB consists of the period of time between the earliest predicted arrival time and the latest predicted departure time, i.e., the latest predicted arrival time plus handling time, where the different predictions are made by the different ML methods applied. That way the uncertainty is considered—following a proactive approach—through a set of predicted arrival times per vessel. While the DTBs of two vessels may overlap, to avoid conflicts between vessels, the berth of one robustly scheduled vessel may not overlap with the DTB or the berth of another vessel.

In order to provide high service quality, vessels' waiting times should be minimized. Therefore, each vessel's berthing time should be scheduled as early as possible. As a vessel cannot be berthed before it arrives at a port, it would be inefficient to allow a berthing time for any vessel prior to its earliest predicted arrival time. Likewise, it is contrary to the planning objective to berth a vessel much later than its latest predicted arrival time (as long as enough capacity is available), as that would lead to a long potential waiting time. Therefore, the earliest and latest arrival time predictions are used as bounds for the planned berthing time. Note that non-robust scheduling and postponement of vessels are also possible in the approach taken in this work, but only if a robust assignment, i.e., an assignment which respects the upper bound, is not possible. This is explained in more detail in Sect. 4.2.

The berthing schedule's robustness is evaluated *ex post*, i.e., after the optimization, when, based on the knowledge of the actual arrival times, actual conflicts can be identified. These arise from the deviation of the actual arrival time from the predictions and lead to infeasibility of the berthing schedule. The proportion of the assigned vessels that can be served without conflicts according to the *actual vessel arrival times* is defined as the *true service level* (Kolley et al. 2021). While the value of this measure is only known *ex post*, it is the best measure for evaluating

and comparing the actual quality of different berthing schedules when real data is considered.

4.2 Robust BAP model with DTBs: model ro-DTB-BAPc

The robust model with DTB for the BAP with continuous quay (ro-DTB-BAPc) which is presented below is based on the “Model 1” by Liu et al. (2017) who, as stated above, also aim at a robust berth allocation, taking uncertainty regarding vessel arrival and handling times into account through a scenario-based approach. In this work, a special kind of scenarios is used; for all relevant vessels the arrival times are predicted by each of the four ML methods, i.e. by LR, kNN, DTR and ANN, as described in Sect. 3, such that four different scenarios result which are simultaneously considered in the ro-DTB-BAPc. Hence, in the approach taken here, ML-for arrival time prediction and scenario development-and optimization-for berth allocation-are combined. The model’s further modifications and enhancements, which were made to improve the robustness of the resultant berthing schedules, are explained in more detail below; among others, they consist of the assignment of a berth to each arriving vessel (no rejections as in “Model 1”) and the consideration of the DTBs which do not exist in “Model 1” by Liu et al. (2017). Additionally, the service level is to be maximized in this work, whereas in the “Model 1” the service level is restricted by a lower bound, i.e. at least a certain degree of robustness is to be achieved. Both models were implemented in Gurobi for Python and solved by the respective standard solver (for the results, see Sect. 5); so both models are used as *solution models* in this study and no specific solution procedures are developed as part of this work.

The ro-DTB-BAPc allocates a known set of vessels V to the available quay space, considering a set of forecasts Ω to predict the respective vessels’ arrival times. Each forecast $\omega \in \Omega$ provides a predicted arrival time $a_i(\omega)$ for each incoming vessel $i \in V$. The weights for each forecast $\omega \in \Omega$ are set to identical values, $p(\omega) = \frac{1}{|\Omega|}$.

The available capacity for assigning the $N = |V|$ vessels is defined by the quay’s length L and the planning horizon T . A berth for a vessel $i \in V$ is assigned according to the predicted arrival times $a_i(\omega)$ for that vessel and the pre-scheduled berthing position b_i . The latter is planned in the pre-scheduling step, where a baseline schedule is developed; this can be any feasible solution of the BAP (Umang et al. 2017).

The presented model aims at maximizing customer satisfaction. Therefore, three aspects of the service quality provided are considered: (1) the vessel waiting time, (2) the deviation from the pre-scheduled berthing position and (3) the non-robust assignments and postponement of vessels, i.e., a high service level. For each of these aspects, service quality penalty costs arise, where the cost rates c_i^1 and c_i^2 indicate the penalties per minute vessel $i \in V$ potentially has to wait for its berth according to the forecasts and per meter of distance between the berth and vessel i ’s pre-scheduled berthing position b_i .

A vessel is to be assigned *non-robustly* whenever, due to a lack of capacity, it cannot be assigned according to its predicted arrival times (hence the DTB is not

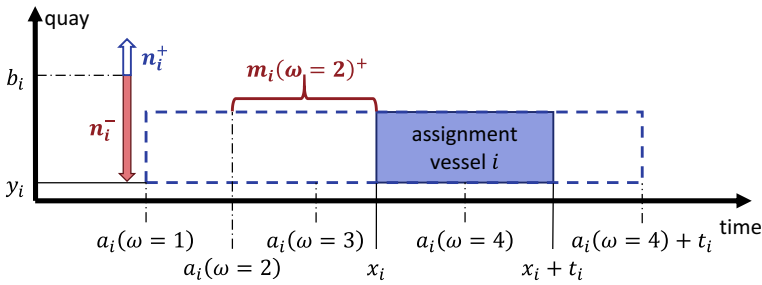


Fig. 5 Illustration of relevant deviation variables

considered) and therefore has to wait, e.g., lying on the roadstead, until a berth is available at a later time.

Moreover, when the vessel is to be assigned to a berth such that the container handling process ends later than the planning horizon T , the vessel is called *postponed*. This means that the berth is shifted to the right in the time–space diagram (see Fig. 10). In both cases, i.e., for non-robust assignments as well as for postponed vessels, high penalty costs c_i^3 arise. Thus, for a vessel that cannot be served robustly due to a lack of capacity at the time it is expected to arrive, penalty costs have to be incurred for each minute of vessel waiting time *and* additionally for being postponed to a much later berth.

The sizes of the different penalty cost rates should be chosen such that they model the relations between the degrees of importance of the three aspects of service quality. In particular, in order to avoid a large number of vessels being postponed, the related penalty cost c_i^3 has to be significantly higher than the other two cost rates.

It should be noted that aiming at the maximization of the terminal’s profit and therefore considering revenue in the objective function, would not influence the solution of the model, as all vessels are to be assigned and hence the revenue from serving the vessels is predetermined. Moreover, only penalty costs are considered, while for profit maximization, e.g., handling costs would have to be taken into account as well.

The possibility of postponing vessels exceeds the capabilities of the original “Model 1” by Liu et al. (2017) where this option is not given; instead, in their model vessels are “rejected” when they cannot not be scheduled without conflict. As an actual rejection is not realistic for a container terminal, this can be interpreted as a postponement to a later, yet unknown point in time. On the long-term all vessels are assigned, even the rejected ones and hence the resulting revenue is predetermined also for their approach. If vessels have to be rejected, these vessels are virtually “stacked” by the “Model 1”. Formally, they are all assigned to a similar time and position within the planning horizon in order to safe quay capacity; this effect is shown in Fig. 9 and explained in more detail below. In contrast, in the approach taken in this work, a conflict-free berthing time and position is assigned to each vessel. Therefore, also postponed vessels occupy capacity, which is not the case in Liu et al.’s virtual “stacking” approach and also is a refinement of the approach developed by Kolley et al. (2021).

The allocation of a berth to vessel $i \in V$ is modelled with the decision variables (start of) berthing time x_i and berthing position y_i , defining the location of the quay at which the bow of the vessel is placed. The berthing time of vessel $i \in V$ can be assigned earlier or later than the predicted arrival time $a_i(\omega)$ (see Fig. 5). Therefore, a potential waiting time $m_i(\omega)^+$ occurs according to the forecast $\omega \in \Omega$, whenever the assigned berthing time starts later than the predicted arrival time. Vice versa, a potential delay $m_i(\omega)^-$ occurs according to the forecast $\omega \in \Omega$, when the berthing time that is assigned starts earlier than the predicted arrival time. The variables n_i^+ (n_i^-) contain the positive (negative) deviation of the berthing position y_i from the pre-scheduled berthing position b_i of vessel $i \in V$ in meters. t_i describes the handling time of vessel $i \in V$. It is assumed that all vessels arrive early enough such that the container handling could be finished before the end of the planning horizon T , given that there is enough capacity.

Different binary variables are used to model (and avoid) a possible overlap between the berths of two vessels $i, j \in V$ in the time dimension, θ_{ij}^x and in the quay space dimension, θ_{ij}^y . These variables take value 1, if the two vessels do not overlap in the respective dimension and they take value 0, if they do. For further details on these variables and their mode of operation see, e.g., Liu et al. (2017).

In the model presented below, conflicts are not only avoided between every two vessels, but also between each vessel and the DTBs of the $N - 1$ other vessels. The binary robust berthing decision variable w_i takes value 1 whenever the DTB of vessel i is respected in the berth scheduling, the vessel is served in the planning horizon and, due to the time buffers, the schedule’s stability is enhanced. When a vessel is non-robustly assigned or postponed, the binary robust berthing decision variable w_i takes value 0.

Now the model formulation of the *ro-DTB-BAPc* can be presented using the notation introduced above:

$$\min \sum_{\omega \in \Omega} \sum_{i \in V} p(\omega) \cdot c_i^1 m_i(\omega)^+ + \sum_{i \in V} c_i^2 \cdot (n_i^+ + n_i^-) + \sum_{i \in V} c_i^3 \cdot (1 - w_i) \tag{1}$$

subject to

$$x_i + t_i \leq T + T \cdot (1 - w_i) \quad \forall i \in V \tag{2}$$

$$y_i + l_i \leq L \quad \forall i \in V \tag{3}$$

$$x_i \geq \min_{\omega} a_i(\omega) \quad \forall i \in V \tag{4}$$

$$x_i + t_i \leq x_j + 2 \cdot T \cdot (1 - \theta_{ij}^x) \quad \forall i, j \in V, i \neq j \tag{5}$$

$$a_i(\omega) + t_i \leq x_j + 2 \cdot T \cdot (2 - \theta_{ij}^x - w_i) \quad \forall i, j \in V, \omega \in \Omega, i \neq j \tag{6}$$

$$x_i + t_i \leq a_j(\omega) + 2 \cdot T \cdot (2 - \theta_{ij}^x - w_j) \quad \forall i, j \in V, \omega \in \Omega, i \neq j \tag{7}$$

$$y_i + l_i \leq y_j + L \cdot (1 - \theta_{ij}^y) \quad \forall i, j \in V, i \neq j \tag{8}$$

$$\theta_{ij}^x + \theta_{ji}^x + \theta_{ij}^y + \theta_{ji}^y \geq 1 \quad \forall i, j \in V, i \neq j \tag{9}$$

$$x_i - a_i(\omega) = m_i(\omega)^+ - m_i(\omega)^- \quad \forall i \in V, \omega \in \Omega \tag{10}$$

$$y_i - b_i = n_i^+ - n_i^- \quad \forall i \in V \tag{11}$$

$$m_i(\omega)^+, m_i(\omega)^-, n_i^+, n_i^-, x_i, y_i \geq 0, \tag{12}$$

$$\theta_{ij}^x, \theta_{ij}^y, u_{ij}, w_i \in \{0, 1\} \quad \forall i, j \in V, i \neq j, \omega \in \Omega$$

The objective (1) is to minimize all penalty costs resulting from the three aspects of service quality: The potential waiting time, the spatial deviation from the pre-scheduled berthing position and the non-robust assignments or postponement of vessels. For each vessel i , the cost rates are set individually such that priorities between vessels can be captured as well.

Constraint set (2) restricts the current planning horizon to which vessels can be assigned. If the end of the scheduled container handling ($x_i + t_i$) of a vessel i exceeds the current planning horizon T , the respective vessel will be postponed to the succeeding planning horizon. Constraints (3) limit the berth assignment to the quay range, where the end of a berth $y_i + l_i$ may not exceed the quay's length L . A vessel i 's berthing time x_i cannot be assigned earlier than its earliest predicted arrival time $\min_{\omega} a_i(\omega)$; this is ensured by constraint set (4).

The constraint sets (5)–(7) prevent the berths of vessel i and j from overlapping in terms of time. Constraints (5) state that the berth of a succeeding vessel j does not overlap with the berth of vessel i , when its berthing time x_j is later than the departure time $x_i + t_i$ of the preceding vessel i . Postponed vessels may be assigned in twice the planning horizon (constraints (2)) and hence the maximum time span

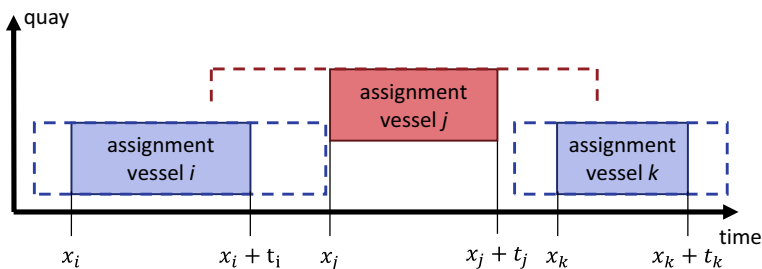


Fig. 6 Robust and non-robust assignments in the berthing schedule

allowed between two vessels' berths is to be set to $2 \cdot T$ as well. Moreover, constraint sets (6) and (7) avoid an overlap of the berth of a vessel with the DTB of another vessel and therefore address the stability of the resulting berthing schedule. If vessel i is scheduled according to its DTB ($w_i = 1$), constraints (6) state that the berthing time x_j of a succeeding vessel j cannot be assigned earlier than the latest predicted departure time of a preceding vessel i (see Fig. 6). Thus vessel j may not overlap with the DTB of vessel i in terms of time. In Fig. 6 it can be seen that the DTB of vessel j cannot be considered in the berth scheduling (illustrated by only the upper half of the rectangle), as it would overlap with the berth assignments of vessel i and k . Thus, vessel j is to be assigned non-robustly and the berthing time x_k of vessel k can be scheduled without considering the latest predicted departure time of vessel j . Likewise, constraints (7) ensure that, without overlapping with vessel j 's DTB, the planned departure time of a vessel i , given by its berthing time x_i plus the handling time t_i , cannot be assigned later than the predicted arrival times $a_j(\omega)$ according to each forecast ω of the succeeding vessel j (and hence not later than the earliest predicted arrival time), if the vessel j will be scheduled according to its DTB ($w_j = 1$).

Note that in Fig. 6 the assignment of vessel j is non-robust and hence its DTB is not respected here. However, in the berth scheduling of vessel j the DTB of vessel i is considered—vessel j is scheduled outside the DTB of vessel i —, as vessel i is robustly scheduled ($w_i = 1$).

Constraint set (8) states that the berths of two vessels i and j do not overlap in terms of space if the berth of vessel i ends on a lower position ($y_i + l_i$) than the berthing position y_j of vessel j . No conflict arises between two vessels i and j , if their berths do not overlap in at least one dimension, i.e., at least one of the binary variables θ_{ij}^x , θ_{ji}^x , θ_{ij}^y or θ_{ji}^y takes value 1 and constraints (9) ensure that this is true for all vessels.

The potential waiting times (or delays) according to the related forecasts ω are determined in constraints (10). Similarly, constraints (11) measure the distance between the assigned berthing position y_i and the pre-scheduled berthing position b_i of a vessel i . Non-negativity and binary conditions for the variables are given in (12).

4.3 Comparison of ro-DTB-BAPc and “model 1” by Liu et al. (2017)

As the “Model 1” proposed by Liu et al. (2017) serves as a basis and a benchmark for the ro-DTB-BAPc model, the major differences between the two models are explained in the following.

“Model 1” contains additional decision variables u_{ij} stating whether two vessels i and j are in conflict ($u_{ij} = 0$), i.e., if their berths overlap or not. Using these decision variables, in “Model 1” the decision is made whether vessels are assigned without conflicts with other vessels. Vessels which are in conflict are virtually “stacked” at a certain berth position and time, i.e., they are assigned to similar berths which necessarily overlap. This effect is due to the fact that the aim of the “Model 1” is the minimization of penalties for waiting times and position deviations, while there is

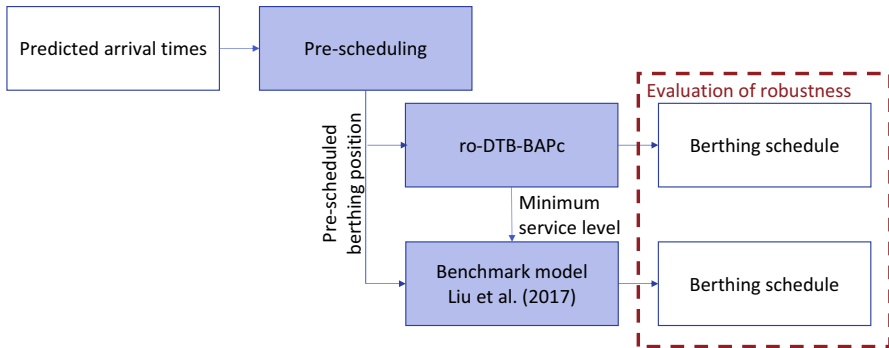


Fig. 7 Optimization and evaluation process

no penalty for vessels which cannot be assigned without conflict as it is the case in the ro-DTB-BAPc. Instead, in “Model 1” an additional constraint requires fulfilment of a certain service level; this is given by a predefined proportion of vessels which have to receive a conflict-free assignment, e.g., 80% of all vessels. Thus, the robustness, measured through the service level, is already specified in the model’s parameters. Hence, in contrast to the model presented above, “Model 1” does not aim at maximizing the berthing schedules robustness, but it only achieves and guarantees a predefined service level.

In the ro-DTB-BAPc, conflicting assignments are not allowed and hence it is ensured by constraints (9) that *all* vessels are scheduled without conflict. Moreover, the DTBs are taken into account, which are not considered by Liu et al. and it is decided if a vessel is robustly assigned; this is modelled by the variables w_i , while the decision variables u_{ij} are not necessary in the ro-DTB-BAPc. The DTB approach is modelled in constraints (6) and (7) which do not exist in “Model 1”; the integration of these individual buffers for increased schedule robustness (see Sect. 4.1) is a major change compared to the original model, where combinations “across scenarios”, as they are defined by the DTBs, are not considered.

Moreover, Liu et al.’s “Model 1” only allows vessels to be assigned in the planning horizon T . Hence, the option of postponing vessels (if necessary) as implemented by constraints (2) is a further important new aspect of the ro-DTB-BAPc, also in comparison to the model presented by Kolley et al. (2021).

Constraint sets (3), (4), (10) and (11) are identical to Liu et al.’s model. The same is true for (5) and (8), despite the fact that t_i , θ_{ij}^x and θ_{ij}^y depend on the respective scenario $\omega \in \Omega$ in their approach.

Finally, in the benchmark model not only the estimated arrival times, but also the handling times depend on the respective scenario $\omega \in \Omega$. Therefore, in “Model 1”, the binary variables θ_{ij}^x and θ_{ij}^y are also defined for each scenario. However, as the focus in this work is on the uncertainty in the arrival times, the vessels’ handling times are assumed to be identical across all scenarios here.

To summarise, in the original model by Liu et al. (2017), vessels which cannot be scheduled without conflict are virtually “stacked” in one position because, when the required service level is fulfilled by conflict-free scheduled vessels, no actual, viable

berthing positions are required (and, hence, determined) for the remaining vessels. In contrast, in the ro-DTB-BAPc non-robustly assigned or postponed vessels are included in the berthing schedule and are allocated to actual, feasible berthing positions. This leads to berthing schedules which are of more practical usefulness as they can be directly implemented in reality.

5 Numerical experiments

5.1 Selection and generation of data

In this section, the set-up of the numerical experiments as well as their results are presented. The major purpose of the numerical studies presented below is to evaluate the quality of the solutions resulting from the ro-DTB-BAPc. In order to do this, the results are compared to those achieved by the “Model 1” proposed by Liu et al. (2017) which, as pointed out above, served as a basis for the ro-DTB-BAPc.

The overall process carried out in the numerical study is shown in Fig. 7. For each vessel, 100 different AIS messages in the time during a port approach are sampled. Arrival time predictions are made by all four ML algorithms and are jointly used as a set of possible arrival times for each individual vessel. Based on these predictions which serve as the different scenarios, the vessels’ DTBs are determined and exploited by the optimization model as explained in Sect. 4.1. However, first a pre-schedule phase is conducted to derive the pre-scheduled berthing positions used in both models and only then the ro-DTB-BAPc is solved. The number of vessels that are assigned to a robust berthing position by the ro-DTB-BAPc, i.e., that are scheduled according to their DTB, then is used as the desired minimum service level for the benchmark model by Liu et al. (2017). The resulting berthing schedules of both models are finally evaluated and compared regarding the true service level using the actual (known) vessel arrival times. The less conflicts occur in a berthing schedule with respect to the actual arrival times of the vessels, the more robust is the respective solution.

In the following analysis, the South Florida Container Terminal SFCT at the Port of Miami is considered for the berth scheduling of incoming container vessels. Overall, a quay length of 1509 m is available, where only the eastern part of the quay (berth 99 to 140) is considered as it is straight and therefore perfectly matches the assumption of a continuous quay layout. The planning horizon T is set to five work days (7.200 min) in order to plan the activities for a whole week.

The set of vessels that are to be assigned in the planning horizon is randomly created from all vessels in the AIS test data set from 1st of July to December 31st, 2020. One AIS message per vessel is randomly chosen from all port approaches included in the test data. An arrival time prediction is made by each of the four ML algorithms for the respective message, such that four predictions are available per vessel.

From the vessels’ AIS data, also the vessels’ lengths are used in the optimization as the vessel’s length represents the demand for quay space during the container

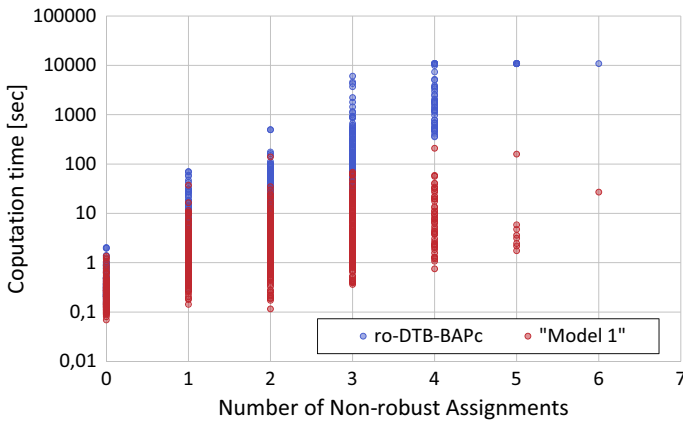


Fig. 8 Computation times of the two optimization models

handling. As the actual handling times are not known, based on a vessel's length, its handling time is generated using normally distributed random variables according to data from the Port of Hamburg⁸ and hence set to realistic values:

For $l < 200m : N(9, 6)$, for $200m \leq l < 300m : N(21, 9)$, for $l \geq 300m : N(32, 8)$.

As the forecasts are originally designed for a period of 24 h ahead, a random offset is added to the predicted arrival time in order to generate a data set with arrival times equally distributed over a five-day period. It is assumed that all vessels arrive at least by the duration of their handling time before the end of the planning horizon, i.e., before the end of the fifth day. Therefore, the offset is chosen such that all vessels could be served within the planning horizon if capacity was available. Note that the actual arrival time with the respective off-set is only used for the retrospective analysis of the berthing schedule; it has no influence on the results of the optimization.

The cost rates for the optimization models depend on the vessels' lengths as well. Hence, bigger deep-sea vessels are prioritised higher than smaller feeder vessels. Because these cost rates are fictional, the actual values are less important than the relation between them to ensure that a postponement of any vessel only takes place when unavoidable ($c_i^2 = 0.2 \cdot c_i^1$, $c_i^3 = c_i^1 \cdot 10^6$).

As the quay's capacity is limited in the planning horizon, the number of vessels that can be assigned to feasible berthing positions is finite as well. Thus, a preliminary test series was conducted to determine a reasonable number of vessels to assign. Three datasets consisting of 20 vessels were created and, starting with a single vessel, in each run one vessel was added to the set of vessels to be scheduled. The solution of the respective ro-DTB-BAPc then gave the planned service level which the benchmark model from Liu et al. (2017) subsequently had to fulfil.

⁸ HHLA (2021): Report Schiffsabfertigung: Segelliste.

URL: <https://coast.hhla.de/report?id=Standard-Report-Segelliste>

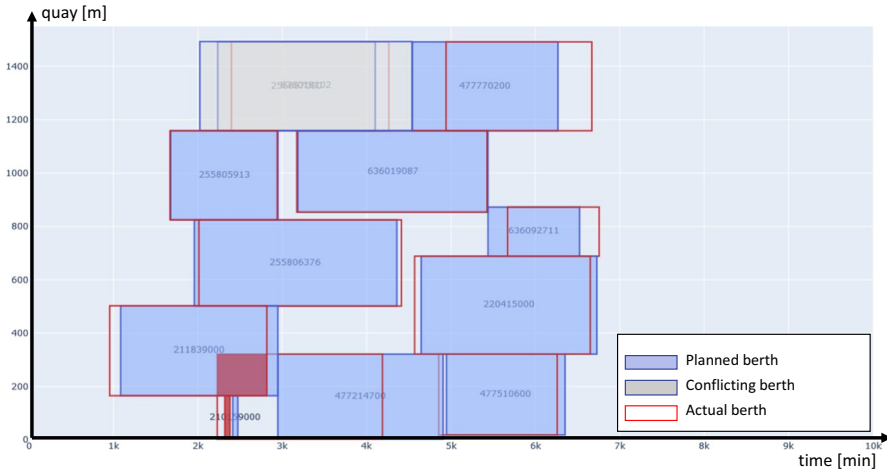


Fig. 9 Resulting berthing schedule from “Model 1”

When considering eleven vessels, in all three datasets at least one vessel was postponed. Hence, not all vessels could be robustly assigned to a berth. However, for one dataset containing twelve vessels, all vessels were assigned robustly. As a suitable dataset size for the available capacity, therefore a maximum number of twelve vessels was chosen according to this preliminary test.

5.2 Solution times and robustness of berth allocation plans

Based on the results from the preliminary test series, for the numerical experiments 1000 datasets were created by randomly choosing twelve vessels per dataset to be scheduled by both models, the ro-DTB-BAPc and the “Model 1” (Liu et al. 2017). The Gurobi library and solver for Python were used for the implementation and the optimization of both models was carried out on a 3.90 GHz 16-core CPU with 64 GB RAM.

The ML forecasting methods used in this work predict the vessels’ arrival times up to 24 h in advance. However, some vessels are much closer to the designated terminal and little time is left until the vessel’s arrival. Thus, the optimization of the berthing schedule is to be conducted in limited time and hence a time limit of three hours is set for solving each model. As shown in Fig. 8, the ro-DTB-BAPc generally takes more computational effort to be solved. However, while “Model 1” determines a schedule and decides which vessels are to be rejected, the ro-DTB-BAPc also makes the decision whether a vessel is robustly assigned according to the DTBs and determines viable berths for the non-robustly scheduled vessels. So, the problem that is solved is more complex, hence solution times should be expected to be longer.

On average, solving the ro-DTB-BAPc with twelve vessels takes 346.57 s, whereas the benchmark model can be solved in 4.71 s on average. The computation time of the ro-DTB-BAPc model increases with a decreasing number of robustly

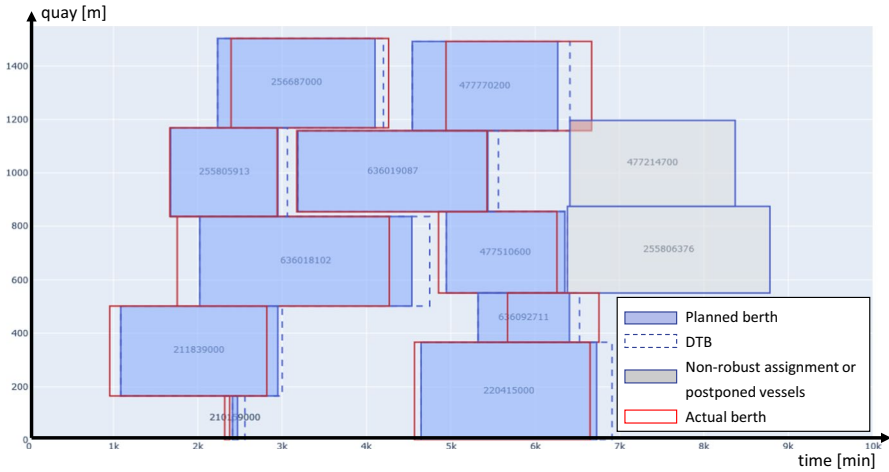


Fig. 10 Resulting berthing schedule from the ro-DTB-BAPc

assigned vessels, i.e., when it is difficult to adhere to the vessels’ DTBs, leading to a lower planned service level and more non-robust assignments. For datasets where only six vessels are robustly assigned (service level of 50%), the ro-DTB-BAPc was never able to be solved within the pre-set time limit. Moreover, when seven vessels are robustly assigned, for some datasets the ro-DTB-BAPc reached the time limit as well.

A large number of non-robust assignments results from a lack of capacity at the quay. Thus, in those cases more (or larger) vessels request service at the container terminal than actually can be handled at a time, i.e., the demand significantly exceeds the capacity. On the other hand, when only few vessels are to be assigned non-robustly (here: not more than two vessels, as shown in Fig. 8), the models’ computation times differ less.

In the following, the measures of service quality are analysed. The ro-DTB-BAPc only creates about 44 min in potential waiting time per schedule on average for the robust assignments according to the predicted arrival times and it creates a total average spatial deviation of about 2006 m per schedule for all assigned vessels. “Model 1” leads to an average potential waiting time of about 375 min for the scheduled vessels (per schedule) and to a total average spatial deviation of about 1451 m. Hence, the ro-DTB-BAPc tends to assign vessels earlier than the benchmark model, but it accepts higher spatial deviations instead. Note that only robustly assigned vessels are compared, as the “Model 1”’s stacking approach does not allow for reasonable evaluation of the non-robustly scheduled vessels.

The ro-DTB-BAPc determines the minimum service level that is to be fulfilled by the “Model 1”. Therefore, there is no difference between the solutions concerning the planned service level. On average, the planned service level is 84.74%, representing 10.17 robustly assigned vessels.

It is worth noting that the “Model 1” by Liu et al. (2017) rejects vessels which cannot be assigned. As explained above, these rejected vessels can be virtually

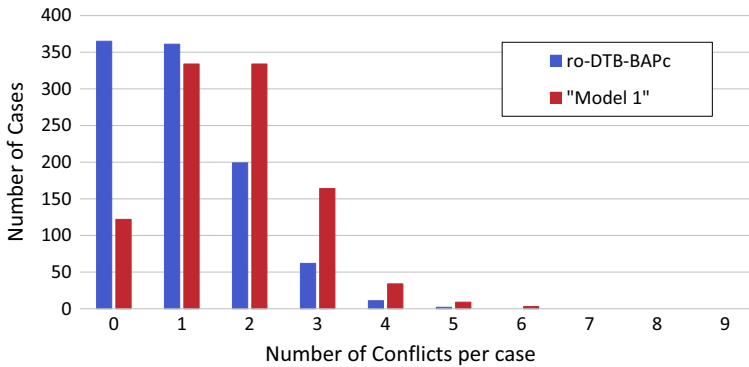


Fig. 11 Distribution of the number of conflicts per data set

“stacked” in order to reduce the resulting penalty costs. An example is given below in Fig. 9, where the rejected vessels are indicated as light blue rectangles in the upper left-hand-side corner. The other blue rectangles show the non-rejected vessels’ planned berthing positions and the red frames show their actual arrival and berthing times. Hence, the schedule also shows two conflicts resulting from the actual vessel arrivals at the bottom left (red areas).

In contrast to “Model 1”, the ro-DTB-BAPc differentiates between robustly scheduled vessels and non-robust assignments and postponed vessels. These vessels are actually scheduled (just at a later time) as well and receive a berthing time and position (see light blue berth assignments in Fig. 10). For the same data as above, the ro-DTB-BAPc results in only one conflict between a robustly scheduled vessel and a postponed vessel (at the top right corner, marked in red).

Apart from the optimization of the planned service quality, a major goal of the presented approach is to enhance the *actual* service quality. This can only be evaluated retrospectively, i.e., when the actual arrival times of the vessels are known. Then the *actual waiting time* measures the period of time by which the vessels arrive earlier than their scheduled berthing time and hence have to wait for their berth.

The ro-DTB-BAPc reaches an average actual waiting time of about 566 min per berthing schedule considering the robustly scheduled vessels, whereas the “Model 1” leads to an actual waiting time of about 902 min on average according to the vessels’ actual arrival times. Thus, both the amount of potential and actual waiting times are significantly lower for the ro-DTB-BAPc than the respective values achieved by the benchmark model. Especially the lower actual waiting time means that the customers actually have to wait less for their berths and hence experience a better service quality.

The feasibility of the resulting berthing schedule depends on how well the respective model handles the uncertainty in the vessels’ arrival times. Therefore, the true service level is considered, which is the ratio of robustly scheduled vessels that can be served without conflict according to their actual arrival time. For the benchmark model solely conflicts between assigned vessels can be evaluated. However, for the

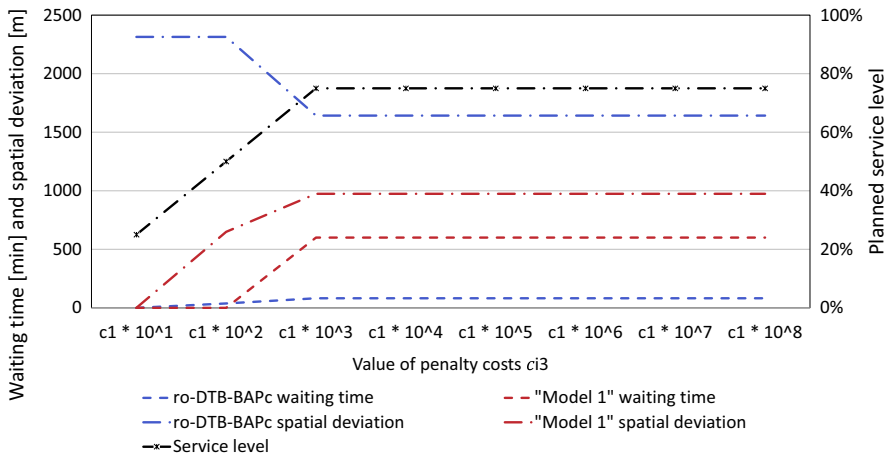


Fig. 12 Influence of size of penalty costs for non-robust assignments and postponed vessels on waiting time, spatial deviation and service level

ro-DTB-BAPc conflicts are analysed between all vessels, as all vessels receive a berth and can be evaluated regarding their actual arrival time.

The ro-DTB-BAPc leads to fully robust berthing schedules in 365 of the 1000 cases and to at most one conflict in 72.6% of the cases (see Fig. 11). Over all 1000 datasets, the ro-DTB-BAPc leads to an average of one conflict per case according to the vessels’ actual arrival times. The resulting true service level is 78.88% on average of the robustly scheduled vessels, considering also conflicts between the robustly scheduled vessels and the non-robust assignments or postponed vessels.

In contrast, the benchmark model from Liu et al. (2017) gives a berthing schedule without conflict in only 122 cases and in 45.6% of the cases at most one conflict arises. On average, “Model 1” results in about 1.7 conflicts per case. Hence, overall the “Model 1” leads to significantly more conflicts per case than the ro-DTB-BAPc with respect to the actual arrival times, providing an average true service level of 65.49% only considering conflicts between the scheduled vessels. Overall, the ro-DTB-BAPc results in an at least equally robust or in an even better (more robust) berthing schedule than “Model 1” in about 85% of the cases.

5.3 Sensitivity analysis for penalty cost parameter c_i^3

A variation of the penalty costs parameter for non-robust assignments and postponed vessels c_i^3 is conducted to determine the influence of the parameter’s value on the resulting berthing schedule. Thus, for a specific dataset the value of c_i^3 is varied in the interval $[c_i^1 \cdot 10^1, c_i^1 \cdot 10^8]$, as shown in Fig. 12. For $c_i^3 \geq c_i^1 \cdot 10^3$, the solution remains the same and higher penalty costs have no influence on the resulting berthing schedule, as a planned service level of 75% cannot be exceeded for

Table 5 Deviation of the berthing time from the actual arrival time per schedule

Average of total ... per schedule	ro-DTB-BAPc	Model 1 (Liu et al. 2017)
Actual waiting time [min]	565.71	902.47
Actual delay [min]	707.40	611.67
Sum of deviations [min]	1,273.11	1,514.14

capacity reasons. Decreasing the penalty costs c_i^3 leads to less robust assignments. For the lowest penalty costs considered, only 25% of the vessels are robustly assigned, as can be seen from the graph for the planned service level (black) in Fig. 12 for $c_i^3 < c_i^1 \cdot 10^3$. Since the aim of the penalty costs c_i^3 is to assure that as many vessels are robustly assigned as capacity is available, c_i^3 should be set to at least $c_i^1 \cdot 10^3$. This makes sense, as the lower the penalty costs for non-robust assignments and postponed vessels are, the less is the incentive to assign vessels robustly. On the other hand, with high penalty costs c_i^3 , more potential waiting time and deviation from the pre-scheduled berthing position are accepted in order to assign an additional vessel robustly instead of making a non-robust assignment. This is the case for both models, but the effect is more intense for the “Model 1”. However, with respect to the spatial deviation from the pre-scheduled berthing positions, the ro-DTB-BAPc does behave differently from the benchmark “Model 1”. For the dataset at hand, the spatial deviation increases with a decreasing service level due to the fact that vessels are not rejected (as in “Model 1”) but are non-robustly assigned, often to positions which deviate significantly from the preschedule.

It is worth noting that the variation of the penalty cost parameter c_i^3 does not directly influence the solution of the “Model 1”. But by variation of c_i^3 the planned service level resulting from the ro-DTB-BAPc solution may vary and hence the minimum service level to be fulfilled by the “Model 1” changes accordingly. Thus, there is only an indirect relation between the penalty cost parameter c_i^3 and the solution of the benchmark model.

From the variation of the penalty costs for non-robust assignments and postponed vessels, c_i^3 , it can be concluded that a sufficient size of the parameter is required to assure a high planned service level. When maximizing the service level and, hence, schedule robustness, is not the main objective and more efficient solutions are searched for, the penalty cost parameter can be reduced. However, for the ro-DTB-BAPc the potential waiting time is already low either way and the spatial deviation from the pre-scheduled berthing position decreases with increasing service level, until the maximum service level is reached. Hence, at least for the data considered in this work, sufficiently high penalty costs c_i^3 result in an advantageous solution with respect to all three aspects of service quality.

5.4 Influence of the prediction accuracy on the berthing schedule

Similar to the actual waiting time, the actual delay marks the period of time by which the robustly scheduled vessels arrive later than their planned berthing time. Each individual vessel increases the actual total waiting time when it arrives earlier and the actual delay when it arrives later than expected. The overall deviation of the actual arrival times from the berthing times of all vessels is the sum of both, the absolute actual waiting time and the absolute actual delay per schedule (see Table 5).

It can be seen that the ro-DTB-BAPc tends to assign vessels earlier than the “Model 1”, as the actual total waiting time is lower, but the actual total delay is higher. This coincides with the lower potential waiting time discussed above. On average, the ro-DTB-BAPc results in about 240 min less deviation of the planned berthing times from the actual arrival times than the benchmark model.

The share of the deviation is 125.18 min per robustly scheduled vessel for ro-DTB-BAPc, while the corresponding share of the deviation per vessel of the “Model 1” is 148.88 min. The meaning of these numbers is equivalent to the MAE from the ML evaluation in Sect. 3.4, stating the mean of the absolute deviations of the forecasts (here the berthing times) from the actual arrival times. Hence, the ro-DTB-BAPc benefits more from the use of the different predictions. Moreover, the accuracy of the resulting berthing schedules exceeds the accuracy of each individual forecast (see MAE entries Table 4), whereas the “Model 1” results in a higher error than the forecasts, except for the LR forecast. It can be concluded that the ro-DTB-BAPc can successfully exploit the information on arrival time uncertainty from the arrival time predictions.

Although time buffers are assigned by the ro-DTB-BAPc, increasing the quay’s idle time, the benchmark “Model 1” nevertheless results on average in a higher sum of the deviations of the actual arrival time from the planned berthing time. In particular, the actual waiting time is much higher than for the ro-DTB-BAPc, indicating that the vessels often are assigned too late in planning horizon such that the “Model 1” results in a more conservative berthing schedule. This leads to a loss in efficiency, as the available time and space at the quay are not fully exploited. So the ro-DTB-BAPc has a clear advantage in this respect.

Therefore, even though it might be suspected that adding time buffers to the berthing schedule to tackle the uncertainty would result in more conservative and less efficient solutions, the numerical experiments show that the concept of the DTB leads to *less* actual waiting times and, hence, to an increase in efficiency.

6 Conclusion and outlook

Container terminals connect maritime ports and the hinterland and therefore are essential for global supply chains. However, the use of AIS data and ML techniques is not yet widely established in container terminal optimization, especially not in combination with OR methods. To the best of the authors’ knowledge, the approach presented in this work—to develop an AIS data based forecast using ML

methods for vessel arrival time prediction at a designated container terminal, which is subsequently used in an optimization model for the robust berth allocation problem, the ro-DTB-BAPc is unique and fills part of the research gap on data-driven decision making for container terminals as identified by Heilig et al. (2020).

For the ML approach taken in this work, the necessary data pre-processing and the selection of reasonable ML algorithms are discussed; then, an optimization of the algorithms' hyper-parameters is carried out. The results of the different forecasting methods show that rather simple approaches, e.g., the k NN algorithm, are able to reach high forecast accuracy in this context. Moreover, in most of the error metrics k NN even outperforms the usually very adaptive ANN algorithm. Hence, it can be concluded that more complex ML methods do not necessarily result in higher prediction accuracy.

It should be noted, however, that the data cleansing procedure was not able to eliminate all of the irrelevant data, e.g., when the AIS status is not correctly set to "moored" while a vessel is stationary at another port, this was not identified.

In order to further enhance prediction accuracy, more detailed information about the port can be considered in the ML, e.g., the pilot guiding process, tug boat activities and the position of roadsteads. Moreover, the current weather conditions influence the vessels' trajectories and speed through crosswinds and sea current. Thus, considering weather data might enhance the accuracy of the forecasts, although this was not found to be the case by Parolas (2016). To overcome the influence of missing and faulty entries in the AIS messages, additional databases on vessel data might be considered.

The ML study further identified that the feature subset '*unscaled_reduced*' outperformed the larger feature sets for all ML algorithms. This raises the question if even better feature subsets exist, i.e., if a different combination of features could be found that leads to higher R^2 values. Future research in this respect could be paired with exploring a wider set of regression methods and for each of these methods a wider range of hyper-parameters might be studied. Further in-depth analyses of the resulting forecasts could deepen the understanding of the importance of features and their interaction. This could also be extended to studies of the effect forecast quality has on the resulting berthing schedules and their robustness.

In order to enhance the berthing schedules' stability and robustness, the concept of DTBs is presented and it is shown how more robust berth allocations can be derived by applying the ro-DTB-BAPc model. The ro-DTB-BAPc is compared to the "Model 1" from Liu et al. (2017) regarding the resulting service quality and the true service level reached in a study consisting of 1000 data sets generated from real vessel data. It turns out that the ro-DTB-BAPc model does not only lead to equally or more robust solutions in 85% of the cases, but also to less potential and actual waiting times for the vessels, indicating a higher service quality. However, this comes at the disadvantages of a larger spatial deviation from the pre-scheduled berthing positions, which is approximately 38% higher than for the benchmark model.

Moreover, all vessels are assigned by the ro-DTB-BAPc model; instead of rejecting vessels which cannot be served without conflict within the planning horizon, as it is the case in the benchmark model, vessels can be either postponed or, if necessary, so-called non-robust assignments are possible. When communicating the planned berthing time to the vessels, for vessels which are to be postponed this can be viewed as an incentive for slow steaming, as they are assigned much later than the predicted arrival times and hence would have to wait for their berth when proceeding at the current speed. Hence, using the proposed method also might enable improvements in energy efficiency.

For future research, deep-learning ML approaches might be employed for the forecasting; as the simpler ML methods led to better results in the studies presented in this work, it would be interesting to see if deep-learning still has an advantage here. Moreover, uncertainty in the vessel handling times could be considered as well. It is to be analysed whether the handling times can be forecasted using ML techniques and the influence on the robustness of the resulting berthing schedules is to be studied. Moreover, the different ML-based forecasts are equally weighted in the optimization model in this work. In the future, it can be evaluated if the accuracy of the vessel arrival time prediction for past data is a good predictor of the future accuracy and hence if more specific weights-i.e., the more successful the method, the larger the weight of the respective forecast-lead to even better results. Finally, instead of solving the models by applying a standard solver as Gurobi, specific (heuristic) solution procedures might be developed in order to enable faster solutions of even bigger problem settings.

Funding Open Access funding enabled and organized by Projekt DEAL. This project was partially funded by the Ministry for Science, Research, Equality and Districts Authority (BWFGB), Hamburg, Germany.

Data availability The raw AIS data which are used for forecasting the ship arrival times are available on the websites of the National Oceanic and Atmospheric Administration (NOAA) Office for Coastal Management. An overview is published here: <https://marinecadastre.gov/ais/>. A refined dataset is available on the TU Hamburg Open Research (TORE) repository under the following <https://doi.org/10.15480/336.4471>. The dataset consists of edited data from the above mentioned raw AIS data. The data is organised as follows: There is one csv file per cargo ship. This file contains all calls of the respective ship at the Port of Miami for the given period from 2018 to 2020. To anonymize the data, the identifying ship data, i.e. the MMSI, was removed. Further information can be found in the paper in Sect. 3.1. The different ML methods which are applied in this work are studied with respect to their forecast quality in order to judge which of them might be most appropriate for vessel arrival time prediction.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is

not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- Akiba T, Sano S, Yanase T, Ohta T, Koyama M (2019) Optuna: a next-generation hyperparameter optimization framework. In: Proceedings of the 25th SIGKDD international conference on knowledge discovery and data mining. ACM, Anchorage, pp 2623–2631
- Arguedas VF, Pallotta G, Vespe M (2018) Maritime traffic networks: from historical positioning data to unsupervised maritime traffic monitoring. *IEEE Trans Intell Transp Syst* 19:722–732. <https://doi.org/10.1109/tits.2017.2699635>
- Baumgarten P, Malina R, Lange A (2014) The impact of hubbing concentration on flight delays within airline networks: an empirical analysis of the US domestic market. *Transp Res E Logist Transp Res* 66:103–114. <https://doi.org/10.1016/j.tre.2014.03.007>
- Belousov B, Abdulsamad H, Klink P, Parisi S, Peters J (Eds) (2021) Reinforcement learning algorithms: analysis and applications. *Studies in computational intelligence*, vol 883. Springer, Cham: Denmark <https://doi.org/10.1007/978-3-030-41188-6>
- Bergstra J, Bardenet R, Bengio Y, Kégl B (2011) Algorithms for hyper-parameter optimization. In: *Advances in neural information processing systems* 24. Curran Associates, Granada. <https://proceedings.neurips.cc/paper/2011/file/86e8f7ab32cfd12577bc2619bc635690-Paper.pdf>
- Bergstra J, Bengio Y (2012) Random search for hyper-parameter optimization. *J Mach Learn Res* 13:281–305
- Bierwirth C, Meisel F (2010) A survey of berth allocation and quay crane scheduling problems in container terminals. *Eur J Oper Res* 202:615–627. <https://doi.org/10.1016/j.ejor.2009.05.031>
- Bierwirth C, Meisel F (2015) A follow-up survey of berth allocation and quay crane scheduling problems in container terminals. *Eur J Oper Res* 244:675–689. <https://doi.org/10.1016/j.ejor.2014.12.030>
- Bose I, Mahapatra RK (2001) Business data mining—a machine learning perspective. *Inf Manag* 39:211–225. [https://doi.org/10.1016/s0378-7206\(01\)00091-x](https://doi.org/10.1016/s0378-7206(01)00091-x)
- Carlo HJ, Vis IFA, Roodbergen KJ (2015) Seaside operations in container terminals: literature overview, trends, and research directions. *Flex Serv Manuf J* 27:224–262. <https://doi.org/10.1007/s10696-013-9178-3>
- Chen M, Hao Y, Hwang K, Wang L, Wang L (2017) Disease prediction by machine learning over big data from healthcare communities. *IEEE Access* 5:8869–8879. <https://doi.org/10.1109/ACCESS.2017.2694446>
- Chen S, Wang F, Wei X, Tan Z, Wang H (2020) Analysis of tugboat activities using AIS data for the Tianjin port. *Transp Res Rec* 2674:498–509. <https://doi.org/10.1177/0361198120916734>
- Cheimanoff N, Fontane F, Kitri MN, Tchernev N (2021) A reduced VNS based approach for the dynamic continuous berth allocation problem in bulk terminals with tidal constraints. *Expert Syst Appl* 168:114215. <https://doi.org/10.1016/j.eswa.2020.114215>
- de León AD, Lalla-Ruiz E, Melián-Batista B, Moreno-Vega JM (2017) A machine learning-based system for berth scheduling at bulk terminals. *Expert Syst Appl* 87:170–182. <https://doi.org/10.1016/j.eswa.2017.06.010>
- Dobrkvic A, Iacob M-E, van Hillegersberg J, Mes MRK, Glandrup M (2016) Towards an approach for long term AIS-based prediction of vessel arrival times. In: Zijm H, Klumpp M, Clause U, ten Hompel M (eds) *Logistics and supply chain innovation: bridging the gap between theory and practice*, 1st edn. Springer, Cham. https://doi.org/10.1007/978-3-319-22288-2_16
- Dobrkvic A, Iacob M-E, van Hillegersberg J (2018) Maritime pattern extraction and route reconstruction from incomplete AIS data. *Int J Data Sci Anal* 5:111–136. <https://doi.org/10.1007/s41060-017-0092-8>
- Dornemann J, Rückert N, Fischer K, Taraz A (2020) Artificial intelligence and operations research in maritime logistics. In: Jahn C, Kersten W, Ringle CM (Eds.) *Data science in maritime and city logistics: data-driven solutions for logistics and sustainability*. Proceedings of the Hamburg international conference of logistics. Epubli GmbH, Hamburg, pp 337–381. <https://doi.org/10.15480/882.3140>

- Eggensperger K, Feurer M, Hutter F, Bergstra K, Snoek J, Hoos HH, Leyton-Brown K (2013) Towards an empirical foundation for assessing bayesian optimization of hyperparameters. In: NIPS Workshop on bayesian optimization in theory and practice. <https://www.cs.ubc.ca/~hoos/Publ/EggEtA113.pdf>
- Fancello G, Pani C, Pisano M, Serra P, Zuddas P, Fadda P (2011) Prediction of arrival times and human resources allocation for container terminal. *Marit Econ Logist* 13:142–173. <https://doi.org/10.1057/mel.2011.3>
- Feng M, Shaw S-L, Peng G, Fang Z (2020) Time efficiency assessment of ship movements in maritime ports: a case study of two ports based on AIS data. *J Transp Geogr* 86:102741. <https://doi.org/10.1016/j.jtrangeo.2020.102741>
- Filom S, Amiri AM, Razavi S (2022) Applications of machine learning methods in port operations—a systematic literature review. *Transp Res E* 161:102722
- Franzkeit J, Pache H, Jahn C (2020) Investigation of vessel waiting times using AIS data. In: Freitag M, Haasis H-D, Kotzab H, Pannek J (eds) *Dynamics in logistics: proceedings of the 7th international conference LDIC 2020*. Springer, Bremen, pp 70–78. https://doi.org/10.1007/978-3-030-44783-0_7
- Frochte J (2019) *Maschinelles Lernen: Grundlagen und algorithmen in Python*, 2nd edn. Hanser, München. <https://doi.org/10.3139/9783446459977>
- Fujino I, Claramunt C, Boudraa A-O (2018) Extracting courses of vessels from AIS data and real-time warning against off-course. In: *Proceedings of the 2nd international conference on big data research*. ACM, Weihai, pp 62–69. <https://doi.org/10.1145/3291801.3291823>
- Géron A (2018) *Praxiseinstieg machine learning Mit Scikit-Learn Und Tensorflow: Konzepte, tools Und Techniken Für Intelligente Systeme*, 1st edn. O'Reilly, Heidelberg
- Grida M, Lee C-Y (2018) An empirical model for estimating berth and sailing times of mega container ships. *Marit Policy Manag* 45:1078–1093. <https://doi.org/10.1080/03088839.2018.1452304v>
- Hendriks M, Laumanns M, Lefeber E, Udding JT (2010) Robust cyclic berth planning of container vessels. *OR Spectr* 32:501–517. <https://doi.org/10.1007/s00291-010-0198-z>
- Heilig L, Stahlbock R, Voß S (2020) From digitalization to data-driven decision making in container terminals. In: Böse JW (ed) *Handbook of terminal planning*, 2nd edn. Springer, Cham, pp 125–154. https://doi.org/10.1007/978-3-030-39990-0_6
- Jahn C, Scheidweiler T (2018) Port call optimization by estimating ships' time of arrival. In: Freitag M, Kotzab H, Pannek J (eds) *Dynamics in logistics: proceedings of the 6th international conference LDIC 2018*. Springer, Bremen, pp 172–177. https://doi.org/10.1007/978-3-319-74225-0_23v
- Kingma DP, Ba JL (2015) ADAM: A method for stochastic optimization. In: *3rd international conference for learning representations*. San Diego. <https://arxiv.org/abs/1412.6980>
- Kolley L, Rückert N, Fischer K (2021) A robust berth allocation optimization procedure based on machine learning. In: Buscher U, Lasch R, Schönberger J (eds) *Logistics management contributions of the section logistics of the German Academic Association for business research*. Springer, Dresden, pp 107–122. https://doi.org/10.1007/978-3-030-85843-8_7
- Kolley L, Rückert N, Kastner M, Jahn C, Fischer K (2022) Robust berth scheduling using machine learning for vessel arrival time prediction-refined dataset. TUHH Universitätsbibliothek. <https://doi.org/10.15480/336.4471>
- Kwon H, Bae H (2021) Prediction of vessel arrival time using auto identification system data. *Int J Innov Comput Inf Control* 17:725–734. <https://doi.org/10.24507/ijicic.17.02.725>
- Langley P, Simon HA (1995) Applications of machine learning and rule induction. *Commun ACM* 38:54–64. <https://doi.org/10.1145/219717.219768>
- Lee H-T, Lee J-S, Son W-J, Cho I-S (2020) Development of machine learning strategy for predicting the risk range of ship's berthing velocity. *J Mar Sci Eng* 8:376. <https://doi.org/10.3390/jmse8050376>
- Leus R, Herroelen W (2007) Scheduling for stability in single-machine production systems. *J Sched* 10:223–235. <https://doi.org/10.1007/s10951-007-0014-z>
- Li Y, Chu F, Zheng F, Kacem I (2019) Integrated berth allocation and quay crane assignment with uncertain maintenance activities. In: Zheng F, Chu F, Liu M (ed) *2019 international conference on industrial engineering and systems management*. IEEE, Shanghai, pp 1–6. <https://doi.org/10.1109/iesm45758.2019.8948115>
- Li B, He Y (2020) Container terminal liner berthing time prediction with computational logistics and deep learning. In: *2020 IEEE international conference on systems, man, and cybernetics*. IEEE, Toronto, pp 2417–2424. <https://doi.org/10.1109/smc42975.2020.9282816>
- Li B, He Y (2021) Computational logistics for container terminal handling systems with deep learning. *Comput Intell Neurosci* 2021:5529914. <https://doi.org/10.1155/2021/5529914>

- Liu DC, Nocedal J (1989) On the limited memory BFGS method for large scale optimization. *Math Prog* 45:503–528. <https://doi.org/10.1007/BF01589116>
- Liu C, Xiang X, Zheng L (2017) Two decision models for berth allocation problem under uncertainty considering service level. *Flex Serv Manuf J* 29:312–344. <https://doi.org/10.1007/s10696-017-9295-5>
- Liu C, Xiang X, Zheng L (2020) A two-stage robust optimization approach for the berth allocation problem under uncertainty. *Flex Serv Manuf J* 32:425–452. <https://doi.org/10.1007/s10696-019-09343-w>
- Mestl T, Dausendschön K (2016) Port ETA Prediction based on AIS Data. In: Bertram V (ed) 15th international conference on computer and it applications in the maritime industries COMPIT 16. Lecce, Italy, pp 331–338.
- Pallotta G, Vespe M, Bryan K (2013) Vessel pattern knowledge discovery from AIS data: a framework for anomaly detection and route prediction. *Entropy* 15:2218–2245. <https://doi.org/10.3390/e15062218>
- Pani C, Fadda P, Fancello G, Frigau L, Mola F (2014) A data mining approach to forecast late arrivals in a transhipment container terminal. *Transport* 29:175–184. <https://doi.org/10.3846/16484142.2014.930714>
- Pani C, Vanelislander T, Fancello G, Cannas M (2015) Prediction of late/early arrivals in container terminals—a qualitative approach. *Eur J Transp Infrastruct Res* 15:536–550. <https://doi.org/10.18757/ejtir.2015.15.4.3096>
- Park K, Sim S, Bae H (2021) Vessel estimated time of arrival prediction system based on a path-finding algorithm. *Marit Transp Res* 2:100012. <https://doi.org/10.1016/j.martra.2021.100012>
- Parolas I (2016) ETA prediction for containerships at the port of Rotterdam using machine learning techniques. MSc Thesis, TU Delft
- Rodriguez-Molins M, Ingolotti L, Barber F, Salido MA, Sierra MR, Puente J (2014a) A genetic algorithm for robust berth allocation and quay crane assignment. *Prog Artif Intell* 2:177–192. <https://doi.org/10.1007/s13748-014-0056-3>
- Rodriguez-Molins M, Salido MA, Barber F (2014b) Robust scheduling for berth allocation and quay crane assignment problem. *Math Probl Eng*. <https://doi.org/10.1155/2014/834927>
- Salcedo-Sanz S, Cornejo Bueno L, Prieto L, Paredes D, García-Herrera R (2018) Feature selection in machine learning prediction for renewable energy applications. *Renew Sustain Energy Rev* 90:728–741. <https://doi.org/10.1016/j.rser.2018.04.008>
- Schmidt-Hieber J (2020) Nonparametric regression using deep neural networks with ReLU activation function. *Ann Stat* 48:1875–1897. <https://doi.org/10.1214/19-AOS1875>
- Scholl A (2001) *Robuste Planung und Optimierung*. Physica-Verlag, Heidelberg
- Shukla AK, Singh P, Vardhan M (2019) A hybrid framework for optimal feature subset selection. *J Intell Fuzzy Syst* 36:2247–2259. <https://doi.org/10.3233/JIFS-169936>
- Takano Y, Miyashiro R (2020) Best subset selection via cross-validation criterion. *TOP* 28:475–488. <https://doi.org/10.1007/s11750-020-00538-1>
- Umang N, Bierlaire M, Ereira AL (2017) Real-time management of berth allocation with stochastic arrival and handling times. *J Sched* 20:67–83. <https://doi.org/10.1007/s10951-016-0480-2>
- Virjonen P, Nevalainen P, Pahikkala T, Heikkonen J (2018) Ship movement prediction using k-NN method. In: 2018 baltic geodetic congress. IEEE. Olsztyn, pp 304–309. <https://doi.org/10.1109/bgc-geomatics.2018.00064>
- Wang Z, Guo C (2018) Minimizing the risk of seaport operations efficiency reduction affected by vessel arrival delay. *Ind Manag Data Syst* 118:1498–1509. <https://doi.org/10.1108/IMDS-12-2017-0563>
- Wu L, Xu Y, Wang F (2020) Identifying port calls of ships by uncertain reasoning with trajectory data. *ISPRS Int J Geo-Inf* 9:756. <https://doi.org/10.3390/ijgi9120756>
- Wu Y, Miao L (2020) A robust scheduling model for continuous berth allocation problem under uncertainty. In: 5th international conference on electromechanical control technology and transportation. IEEE. Nanchang, pp 43–49. <https://doi.org/10.1109/icectt50890.2020.00017>
- Xiang X, Liu C, Miao L (2017) A bi-objective robust model for berth allocation scheduling under uncertainty. *Transp Res E Logist Transp Rev* 106:294–319. <https://doi.org/10.1016/j.tre.2017.07.006>
- Xiao Z, Ponnambalam L, Fu X, Zhang W (2017) Maritime traffic probabilistic forecasting based on vessels' waterway patterns and motion behaviors. *IEEE Trans Intell Transp Syst* 18:3122–3134. <https://doi.org/10.1109/tits.2017.2681810>
- Xu Y, Chen Q, Quan X (2012) Robust berth scheduling with uncertain vessel delay and handling time. *Ann Oper Res* 192:123–140. <https://doi.org/10.1007/s10479-010-0820-0>

- Yang D, Wu L, Wang S, Jia H, Li KX (2019) How big data enriches maritime research—a critical review of automatic identification system (AIS) data applications. *Transp Rev* 39:755–773. <https://doi.org/10.1080/01441647.2019.1649315>
- Yu J, Tang G, Song X, Yu X, Qi Y, Li Da, Zhang Y (2018) Ship arrival prediction and its value on daily container terminal operation. *Ocean Eng* 157:73–86. <https://doi.org/10.1016/j.oceaneng.2018.03.038>
- Zhang G, Patuwu BE, Hu MY (1998) Forecasting with artificial neural networks: the state of the art. *Int J Forecast* 14:35–62. [https://doi.org/10.1016/S0169-2070\(97\)00044-7](https://doi.org/10.1016/S0169-2070(97)00044-7)
- Zhang X, Sun B, Sun J, Gou Z (2014) The berth and quay cranes integrated scheduling based on redundancy policy. In: Proceedings of the 33rd Chinese control conference. IEEE, Nanjing, pp 7595–7600. <https://doi.org/10.1109/chicc.2014.6896265>
- Zhen L, Lee LH, Chew EP (2011) A decision model for berth allocation under uncertainty. *Eur J Oper Res* 212:54–68. <https://doi.org/10.1016/j.ejor.2011.01.021>
- Zheng Y (2015) Trajectory data mining: an overview. *ACM Trans Intell Syst Technol* 6:1–41. <https://doi.org/10.1145/2743025>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Lorenz Kolley received a M.Sc. degree in Logistics, Mobility and Infrastructure from the Hamburg University of Technology (TUHH), Hamburg, Germany, in 2020. After graduating, he began working as a research assistant and Ph.D. candidate at the Institute for Operations Research and Information Systems (ORIS) at the TUHH. Lorenz focuses on the application of machine learning (ML) and optimization methods in the context of maritime logistics.

Nicolas Rückert received a M.Sc. degree in Logistics, Mobility and Infrastructure from the Hamburg University of Technology (TUHH), Hamburg, Germany, in 2019. After graduating, he began working as a research assistant and Ph.D. candidate at the Institute for Operations Research and Information Systems (ORIS) at the TUHH. His research interests include optimization methods in the context of the maritime logistics with focus on Hinterland transportation and horizontal cooperation.

Marvin Kastner received a M.Sc. degree in Informatics from the Hamburg University (UHH), Hamburg, Germany, in 2017. Shortly after graduation, he began working as a research associate and Ph.D. candidate at the Institute of Maritime Logistics that is part of the Hamburg University of Technology (TUHH). Marvin works on machine learning and simulation-based optimization in the maritime supply chain with a focus on tactical and strategic terminal planning activities at container terminals.

Carlos Jahn is Head of the Institute of Maritime Logistics at Hamburg University of Technology as well as of the Fraunhofer Center for Maritime Logistics and Services in Hamburg, Germany. At the beginning of his professional career, Carlos Jahn served as a sailor and later on as an officer in the German Navy. He studied mechanical engineering at the Helmut Schmidt University in Hamburg and economics at the University of Hagen. As a research assistant at the Fraunhofer IFF, Carlos Jahn obtained a doctorate at the Otto von Guericke University of Magdeburg. Carlos Jahn held various management and staff functions in science and industry. In 2009, he accepted the offer of a full professorship at Hamburg University of Technology and was appointed Head of the Institute of Maritime Logistics. The Fraunhofer Center for Maritime Logistics and Services was founded in 2010.

Kathrin Fischer received a Diploma in Mathematics, a Ph.D. degree in Management Science and a Habilitation in Business and Management from the University of Hamburg, Germany. Afterwards, she held a lectureship for Management Science at Aston Business School, Birmingham, UK, and professorships for Logistics in Hamburg and Vienna. Since 2009, she has been a Professor for Operations Research and Information Systems and Head of the Institute for Operations Research and Information Systems at the Hamburg University of Technology. Her major research interests are in the fields of maritime and humanitarian logistics, revenue management and location planning. She published articles in many OR-oriented journals as, e.g., *OR Spectrum*, *Transportation Science*, *Annals of OR* and others.