Predicting train dwell times using XGBoost and simple timetable features for German railways

Theo Döllmann

Angaben zur Veröffentlichung / Publication details:

Döllmann, Theo. 2025. “Predicting train dwell times using XGBoost and simple timetable features for German railways.” Augsburg: Universität Augsburg.

Seminar

seminar@ML

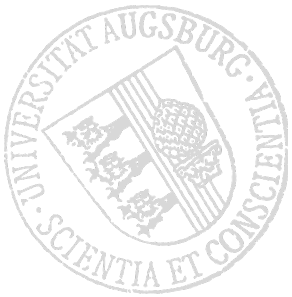
Sommersemester 2025

Predicting Train Dwell Times Using XGBoost and Simple Timetable Features for German Railways

Theo Döllmann
theodor.doellmann@uni-a.de

Advisor: Adrian Pfleiderer
Software Methodologies for Distributed Systems (Prof. Bauer)
University of Augsburg

Abstract To address the critical challenge of railway punctuality, this study develops a stochastic model for predicting train dwell time delays across the German railway network. Using a large-scale dataset of train operations, an XGBoost model is trained on timetable and geographical features to predict a probability distribution over discrete delay intervals. The model's performance is rigorously evaluated against statistical baselines, demonstrating superior accuracy with a Mean Absolute Error of approximately 0.48 minutes. This predictive advantage is shown to be robust over a one-year evaluation period. While the model underestimates peak rush-hour delays, it provides a feasible foundation for a nationwide prediction system.



Contents

1 Introduction

Rail transport is a cornerstone of sustainable mobility, yet its effectiveness is frequently undermined by a lack of punctuality. The German railway network, operated primarily by Deutsche Bahn (DB), faces significant reliability challenges. In April 2025, only 61.9% of DB's long-distance trains and 89.8% of its regional trains arrived with a delay of less than six minutes [1]. Such low punctuality figures not only frustrate passengers and erode confidence in rail travel but also create cascading operational problems for the railway operator.

A reliable train delay forecast could help the DB to optimize planning and could help passengers to make more informed mobility decisions. Most notably, knowing the reliability of a train connection in advance might be very beneficial for customers, as they could avoid unreliable connections.

This study addresses the prediction of train dwell times in the German railway network as a foundational component for a comprehensive delay forecasting system. To use this forecasting system as part of a connection reliability rating system, all of its components must be stochastic, not deterministic. A deterministic component could not be used to give a train connection a stochastic reliability rating.

In the subsequent sections, this paper will first elaborate on the motivation to develop a new prediction method based on a chain-of-models framework. Following this, it will review related work in the field and delineate the specific contributions of this study in comparison to existing methodologies. Section 2 will detail the dataset employed and systematically construct the machine learning model. This includes defining the target variable, outlining the data splitting strategy, explaining the approach to predicting probability distributions, and describing the model selection process, feature set, and hyperparameter tuning methodology. Section 3 will present various evaluation metrics for the delay prediction model, including well-known error measures for regression tasks observed over a one-year period, an analysis of model accuracy across different times of the day, and an evaluation of the predicted probability distributions. All of these evaluations are then discussed in Section 4. Finally, Section 6 will summarize the findings and discuss areas for future research.

1.1 Motivation: Path Aware Chain of Models & Related Work

According to some experiments with train delay data, delays tend to accumulate as a function of the distance or time traveled by a train. Therefore, we hypothesize that a model's performance improves if it is aware of the path traveled by the train. To achieve this, this work proposes a modular framework that chains together predictions from three specialized models: one for origin departure delay, one for running time, and one for dwell time. The prediction for any point in a journey is an aggregation of the outputs from all prior models in the chain.

As an initial step towards this framework, this study focuses exclusively on the prediction of dwell times. In the next few paragraphs, an overview on train delay prediction approaches is given and three dwell time prediction papers are discussed in a

little more detail.

Research on train delay prediction is an active field. Spanninger et al. [2, p. 1] provide a review of various approaches, categorizing them based on their underlying modeling paradigms: event-driven or data-driven. They conclude that data-driven approaches might be more accurate [2, p. 1]. Models discussed in their review exhibit prediction horizons of up to eight hours [2, p. 12].

Kecman and Goverde [3] utilized historical track occupation data to predict running and dwell times on the Leiden-The Hague-Rotterdam-Dordrecht corridor in the Netherlands [3, pp. 1, 299]. Their methodology involved LTS robust linear regression, regression trees, and random forests. They trained both a global model and localized models per line and station, finding that the local models yielded superior performance.

Li, Daamen, and Goverde [4] also employed track occupation data to predict dwell times at one station with a short scheduled dwell time in the Netherlands. Notably, they omitted the number of boarding passengers as a feature, due to its unavailability during inference [4, pp. 1, 11]. Dwell times during peak and off-peak hours were predicted separately using a linear regression model and a k-nearest neighbor model, respectively. They reported an error of less than 10 seconds during peak hours [4, p. 18].

Pang et al. [5] proposed an approach that averages multiple delay predictions and subsequently applies dynamic updating based on exponential smoothing. Their model was evaluated on one high-speed train line in China. According to Pang et al. [5], the updating mechanism reduced the model's error by approximately 15%. For the averaging models, they reported a Mean Absolute Error (MAE) of 0.665 minutes [5, p. 363].

1.2 Contribution

This study investigates the feasibility of predicting train dwell times as a foundational component for a future chain-of-models approach to train delay estimation. In contrast to other studies on this topic, this research aims to develop predictions that are applicable to all trains operating within Germany and that provide stochastic outputs. To determine feasibility, the following criteria must be met: 1. Predictions must exhibit a smaller error than simple statistical baselines, both for the predicted distributions and a derived point prediction from the distribution. 2. Prediction quality must not significantly deteriorate over a prediction horizon of several weeks.

2 Methodology

In this section, the methodological framework employed to develop a stochastic prediction model for train dwell time delays is presented.

2.1 Dataset & Verification

The dataset for this paper was collected from the DB timetable API. Data was gathered by querying all known train stations in Germany for their schedules and real-time stop

updates. For each stop, the most recent timestamp for an arrival or departure event is considered the actual time.

The resulting dataset contains both static timetable information and real-time delay updates for all stops of trains operating in Germany. Each arrival or departure event at a station constitutes a single data point. Each data point includes: general trip information (e.g., train number, operator, category), timetable information (e.g., stop name, planned time, distance traveled), and real-time information (e.g., delay, cancellations). Temporal information has a resolution of one minute. For this paper, 1.18 billion data points collected between December 1, 2021, and December 31, 2024, are used. This period spans three full timetable cycles, as the annual schedule change occurs in mid-December.

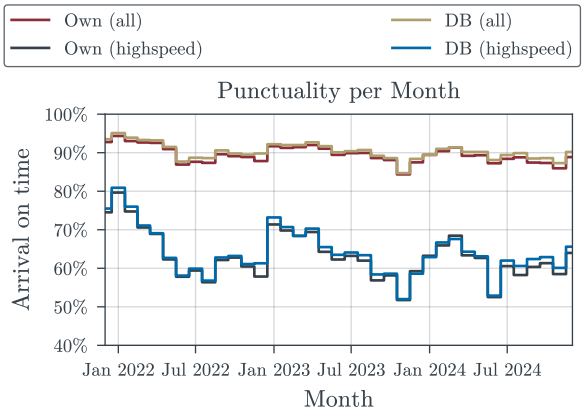


Figure 1: Comparison between official punctuality statistics of DB and statistics build on the dataset.

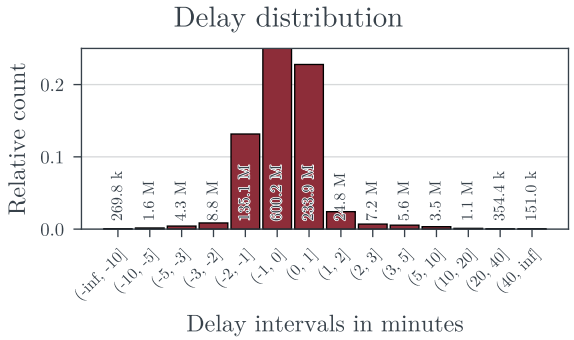


Figure 2: Dwell time delay distribution in the dataset, absolute counts written out. Y-axis clipped at 0.25.

DB provides monthly punctuality statistics, including the percentage of all trains and high-speed trains that are *on time*. A train is considered *on time* if it arrives with a delay of five minutes or less; cancellations are excluded from this metric. As the official statistics website of DB unpublishes numbers older than one year, the numbers were collected as part of a git repository [6]. Every few months, the new punctuality statistics are added to the repository. [1]

Applying this definition to the dataset allows for a comparative validation, as shown in Figure 1. Minor discrepancies between the derived statistics and the official figures are expected, as our dataset is a third party observation. Overall, the dataset shows an average monthly difference of 0.80 % from the official statistics. For high-speed trains, the average difference is 0.99 %. The largest discrepancy occurs in December 2022, where the difference is 1.97 % for all trains and 3.44 % for high-speed trains. This level of accuracy is deemed sufficient for the subsequent machine learning experiments.

2.2 Defining Dwell Time Delay

To predict dwell times, it is first necessary to establish a suitable target variable. For a given train stop, the *scheduled dwell time* (d) is the difference between the scheduled departure (t_{dp}) and arrival (t_{ar}) times:

$$d := t_{\text{dp}} - t_{\text{ar}}$$

Correspondingly, the *actual dwell time* ($\hat{d}$) is the difference between the actual times ($\hat{t}_{\text{dp/ar}}$):

$$\hat{d} := \hat{t}_{\text{dp}} - \hat{t}_{\text{ar}}$$

Instead of predicting the probability distribution for $\hat{d}$, the *dwell time delay* is introduced as the prediction target, defined as the difference between the actual and scheduled dwell times:

$$\text{dwell time delay} := \hat{d} - d$$

From here on, *dwell time delay* will simply be called *delay* for readability. The *delay* is centered around zero, which simplifies the modeling task. As illustrated in Figure 2, most trains have a *delay* of zero minutes. However in absolute numbers, there are still many trains having a delay of more than 40 minutes or less than -10 minutes. Knowing this distribution will be important for building the machine learning model.

2.3 Train and Test Data Splitting

When handling time-series data, it is crucial to design the training and testing split to reflect a real-world environment [7, p. 247]. Predicting the delay for a future period is a typical use case. Consequently, a temporal split for model evaluation is employed with a cutoff point at time t. The training set consists of 27 weeks of data preceding t, while the test set comprises the three weeks of data following t. To manage GPU memory constraints during training, the training data is divided into three sequential nine-week batches. The model is trained incrementally on these batches, starting with the oldest.

2.4 Predicting Probability Distributions

Several methods exist for predicting probability distributions, as described by Tyralis and Papacharalampous [8]. A typical modern approach is to model the parameters of a known distribution function [8, p. 23]. For the problem however, the capability of many classification algorithms to output class probabilities during inference can be exploited, as the *delay* target variable is inherently discrete due to the one-minute resolution. This approach requires defining a finite set of classes by setting lower and upper bounds for the target variable.

Referring to Figure 2, predicting the entire observed range of *delay* is unfeasible. In the context of train delay, huge delays are typically of random nature and not explainable by the features used in this study. Reasons might be staff shortages, extreme weather

or defective infrastructure or rolling stock. For this reason, this study focuses on less extreme cases. We define a prediction range for *delay* from -5 to +9 minutes, inclusive. This creates 15 discrete classes. This interval captures 99.70% of the observed cases, with approximately 0.15% of data being trimmed from each tail of the distribution. The coverage is considered sufficient for this study.

2.5 Model Selection

The selection of an appropriate machine learning model is a critical step. Given the tabular data, which contains a mix of numerical and categorical features, a model adept at handling such structures is required. A review of related work on railway delay prediction (summarized in Figure 3) shows that many studies use deterministic methods for single-step prediction. Queueing models are out of scope as they demand precise network infrastructure models [9, p. 104], and linear regression is ill-suited for stochastic predictions.

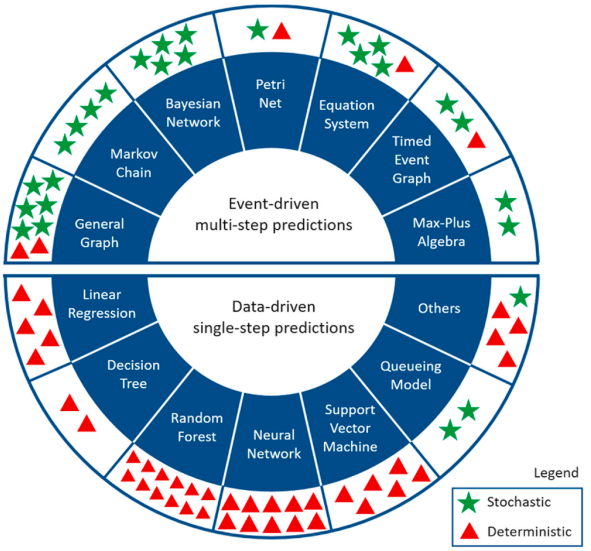


Figure 3: Classification of train delay prediction approaches by Spanninger et al. [2, p. 4]

In prior work related to predicting train delays, both decision trees [10, p. 10] and random forests [11, p. 9] yielded inferior performance compared to XGBoost for train delay prediction tasks. Although undocumented, preliminary experiments with deep neural networks and support vector machines also produced unsatisfactory results. Based on this experience, XGBoost has consistently shown superior performance on the dataset. As the prediction of the *dwell time delay* is conceptually similar to previous arrival and departure delay prediction tasks, XGBoost was selected for this study.

XGBoost has numerous hyperparameters that require careful configuration.

- **Objective**: To align with the goal of predicting a probability distribution, the default `multi:softmax` objective is selected. Other available objectives only differ in the output format.
- **Evaluation Metric**: Ideally, the evaluation metric would account for the ordinal relationship between the classes (e.g., predicting a *delay* of +1 when the truth is +2 is better than predicting -3). However, all standard metrics for multi-class classification assume independent classes [12]. Therefore, the default multi-class logarithmic loss (`mlogloss`) is used, which is the negative log-likelihood.
- **Other Settings**: Categorical feature support and CUDA were enabled.

Further parameters are tuned as described in Section 2.7.

2.6 Simple Timetable Features

For this initial investigation, set of relatively simple features extracted from timetable data was selected. This feature set was adapted from another feature set used in previous train delay prediction tasks [13, pp. 5–6]. An automated feature selection process, described in Section 2.7, is used to eliminate uninformative features from this initial set. The proposed features are detailed in Table 1.

Table 1: Proposed features for the machine learning model. Longitude and Weekday were excluded after feature selection in Section 2.7

Feature name	Feature definition
Train number	The official number identifying the train trip.
Latitude	Latitude of the stop (EPSG:4326).
Longitude	Longitude of the stop (EPSG:4326).
Stop sequence	The sequential index of the stop within the train's trip, starting from 1.
Distance traveled	The cumulative distance traveled by the train from its origin station in meters, calculated using Dijkstra's shortest path algorithm on the OpenStreetMap railway network.
Dwell time	Scheduled dwell time as defined in Section 2.2.
Bearing	True bearing angle between the trip's origin and destination stops. See Bowditch [14, pp. 8–9] for definition.
Minute of day	The number of minutes past midnight for the scheduled arrival, in the local timezone and accounting for daylight saving time.
Weekday	Weekday number of the scheduled arrival.
Is regional	A flag indicating if the train is a regional service.
Operator	Internal short code for the train operating company (e.g. Flixtrain).
Category	The category of the train (e.g., ICE, IC, RE, RB, S).
Line	The line number of the train service (e.g., 9 for RE 9, 1 for S 1). Line numbers are unique and may be reused by different routes.

The `operator`, `category`, and `line` features are treated as categorical, while all others are numeric. Although the `train number` could be considered categorical, its high cardinality makes this computationally infeasible. It is therefore treated as a numeric feature. While there is no direct ordinal relationship, the numbering system contains implicit information (e.g., high-speed trains typically have numbers below 1000) that the model may exploit.

2.7 Hyperparameter Tuning & Feature Selection

Hyperparameter tuning is particularly important for a model as complex as XGBoost. Given the large number of tunable parameters (the documentation lists 24 for the tree

booster [12]), an exhaustive grid search is computationally prohibitive. Based on a review of existing literature [15, p. 5] [16, p. 6557] [17, p. 6] and the official documentation, a subset of key hyperparameters for optimization were selected. The full list of tuned hyperparameters can be found in Table 2. For an explanation of the parameters, please consult the XGBoost parameter manual ([12]).

The black-box optimization framework *Optuna* [18] is used to perform the tuning. *Optuna* can optimize any objective function that returns a single numeric value to be minimized. The ranges, adapted from previous experiments, of the parameter search space for *Optuna* are depicted in Table 2. To integrate the feature selection directly into the tuning process, *Optuna* is allowed to select a subset of the proposed features in each trial. The objective function for Optuna is set to minimize the Root Mean Squared Error (RMSE), calculated between the true integer-encoded *delay* and the expected value of the predicted probability distribution. For concise definitions of the expected value and the RMSE, please consult Section 3 and Section 3.2. The optimization was run for 200 trials.

3 Evaluation

To provide an initial overview of the model's performance, the probabilistic nature of the predictions is initially disregarded for a better interpretability and comparability of the results. Instead, the evaluation focuses on a point estimate derived from the predicted distribution: the expected value. For a predicted probability distribution p over a set of discrete classes $C = \{-5, -4, \ldots, +9\}$, a point estimate is derived by calculating its expected value, $\bar{p}$:

$$\bar{p} := \sum_{c \in C} c \cdot p_c$$

where p_c is the predicted probability for the delay class c. In the subsequent sections, this expected value $\bar{p}$ is used as the point prediction for evaluation against standard regression metrics. For all evaluations, October 1, 2023, was chosen as the training cutoff date t. Unless stated otherwise, the test set comprises the subsequent 21 days (three weeks).

3.1 Baseline Definitions

Evaluating a model's performance score in isolation is of limited value, as the score lacks context. Its significance is only revealed when compared against appropriate baselines. Therefore, three simple statistical baselines that a proficient model should surpass were defined:

- **Average (μ)**: The mean *delay* across all stops in the training data.
- **Average per category ($\mu_{\mathbf{cat}}$)**: The mean *delay* calculated independently for each train category. This baseline accounts for potential systematic differences between train types (e.g., long-distance vs. regional).

- **Average per stop (μ_{stop})**: The mean *delay* calculated for each unique station. This baseline captures location-specific effects.

All baselines are derived from the same training data as the machine learning model.

3.2 Comparison of XGBoost and Baselines

The performance of the XGBoost model is compared to the baselines using the Mean Absolute Error (MAE) and the Root Mean Squared Error (RMSE). Given a vector of predicted values $\hat{y} \in \mathbb{Q}^n$ and a vector of target values $y \in \mathbb{Q}^n$, where n is the number of data points, the MAE and RMSE are defined as:

$$\text{MAE} := \frac{1}{n} \sum_{i=1}^{n} |\hat{y}_i - y_i|$$

$$\text{RMSE} := \sqrt{\frac{1}{n} \sum_{i=1}^{n} (\hat{y}_i - y_i)^2}$$

This comparison is applied to the 21 testing days. Another key question is how the model's performance degrades over a longer prediction horizon. To investigate this, an extended evaluation period of one year following the training cutoff date is applied. The evaluation is conducted on a weekly basis, with performance metrics calculated for each of the 52 subsequent weeks separately.

3.3 Accuracy During the Course of the Day

To conclude the analysis of the point predictions, the model prediction is compared to the temporal pattern over the course of the day. This should show whether the model has captured the influence of the `minute_of_day` feature. For this analysis, both the predicted expected delay ($\bar{p}$) and the true delay are aggregated into 20-minute bins based on the time of day.

3.4 Quality of the Probability Distribution

The evaluation has thus far focused on point estimates derived from the model's probabilistic output. To assess the quality of the predicted distributions themselves, the concept of the *stochastic error distance* (SED) from Diebold and Shin [19] is adapted for discrete distributions. The SED measures the discrepancy between a predicted cumulative distribution function (CDF) and the empirical CDF of the outcome. For the discrete case, we define the SED as the sum of absolute differences between the predicted CDF, F, and the empirical CDF, F^*, at each possible outcome e in the class set C. The empirical CDF is a step function defined as

$$F^*(e) := \begin{cases} 0 & \text{if } e < \hat{y} \\ 1 & \text{if } e \geq \hat{y} \end{cases},$$

where $\hat{y}$ is the observed outcome. The SED is therefore:

$$\mathrm{SED}(F, \hat{y}) := \sum_{e \in C} |F(e) - F^*(e)|.$$

The mean SED of the XGBoost model is once again compared to the three baselines, but the baselines are adjusted and are no longer based on the mean observed delay. Rather, the empirical distribution of delays observed in the training data (either globally, per category, or per stop) is used.

4 Results

This section begins with the results of the hyperparameter tuning and feature selection used as basis for the experimental evaluation of the XGBoost model, which fills up the rest of this section.

4.1 Tuned Hyperparameters & Selected Features

Table 2 shows the tuned hyperparameters alongside the optimal values identified by the tuning process. The joint tuning and feature selection process determined that the `longitude` and `weekday` features were uninformative; they were therefore excluded from the final model. The optimal hyperparameter values found by Optuna are used for all subsequent experiments.

Table 2: Tuning range and resulting optimal value for each hyperparameter.

Parameter	**Tuning range**	**Result**
`n_estimators`	50 - 200	192
`lambda`	1×10^{-8} - 1	2.162×10^{-8}
`alpha`	1×10^{-8} - 1	0.035
`subsample`	0.100 - 1	0.927
`colsample_bytree`	0.100 - 1	0.378
`max_depth`	3 - 14	14
`min_child_weight`	2 - 15	10
`eta`	1×10^{-8} - 1	0.343
`gamma`	1×10^{-8} - 1	0.036

4.2 Comparison of XGBoost and Baselines

The XGBoost model was compared against the baselines. The results presented in Figure 4, show that the XGBoost model achieves the best performance on both metrics. For the MAE, the XGBoost model has an error of 0.478 minutes, outperforming the per-stop baseline (0.558 minutes), the per-category baseline (0.614 minutes), and the

global mean baseline (0.621 minutes). The RMSE results show a similar ranking: the XGBoost model scores 0.805 minutes, followed by the per-stop baseline (0.903 minutes), and both the mean and per-category baselines (0.960 and 0.963 minutes).

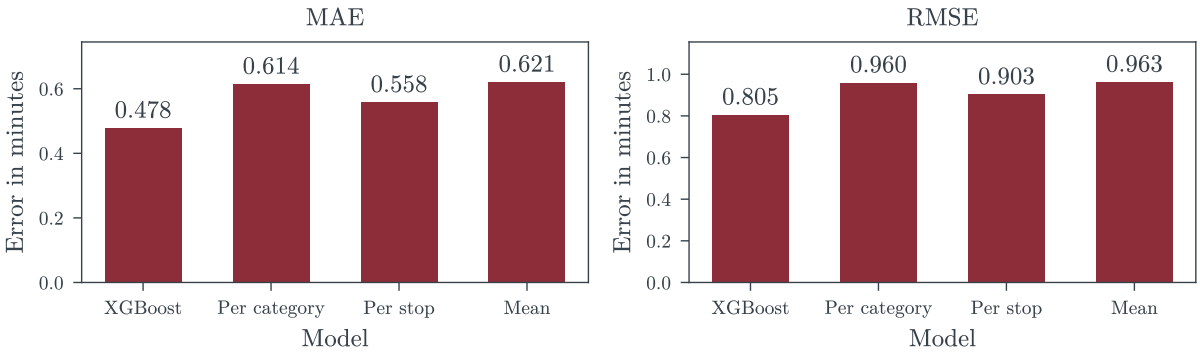


Figure 4: Average prediction accuracy of the XGBoost model compared to three baselines.

The results over the one-year period are shown in Figure 5. The performance ranking is consistent with the initial three-week evaluation: the XGBoost model is the most accurate, followed by the per-stop baseline, and then the remaining two baselines. The performance of all models fluctuates weekly. A notable drop in error for all models occurs in calendar week 50 of 2023. This coincides with the annual timetable change. In week 8 and 20 are two drops in error for all models. Aside from the drops, the error for all models exhibits a weak gradual upward trend over the year.

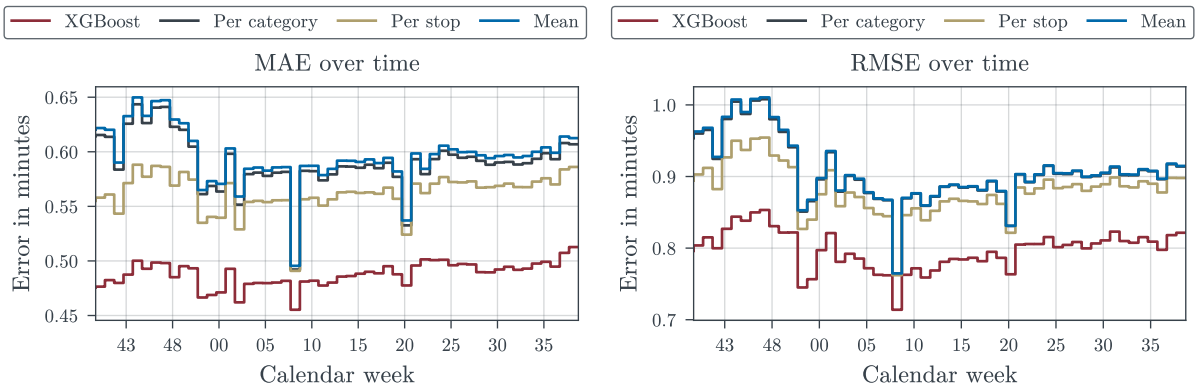


Figure 5: Average prediction accuracy of the XGBoost model compared to three baselines, evaluated weekly from October 1, 2023, to October 1, 2024.

4.3 Accuracy During the Course of the Day

As shown in Figure 6, the observed mean delay (the red curve) exhibits distinct peaks corresponding to morning and evening rush hours, specifically around 07:00, and 17:00, plus a third peak at 04:00, that has no indication of being linked to the activity in the railway network. The XGBoost model's predictions (the black curve) generally follow the trend of the observed data but notably fail to capture the magnitude of the 07:00 peak.

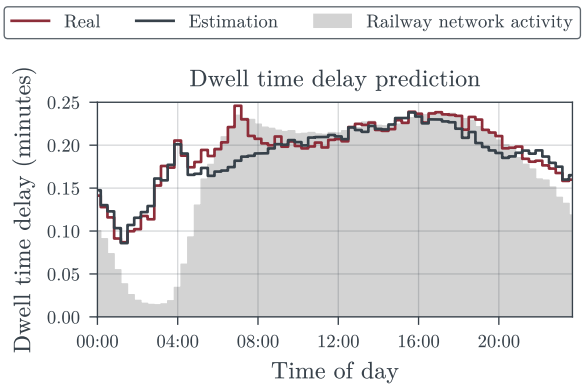


Figure 6: Mean predicted and mean observed delay over the day, aggregated in 20-minute bins.

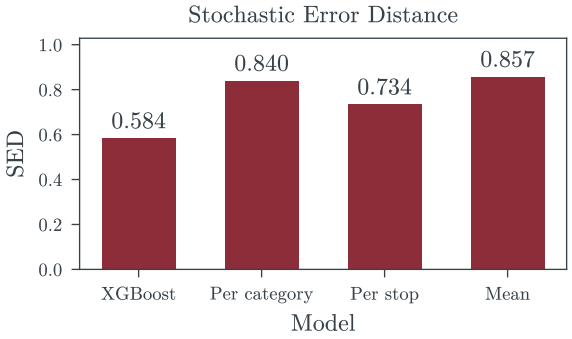


Figure 7: XGBoost SED compared to baselines

4.4 Quality of the Probability Distribution

The results, shown in Figure 7, mirror the findings from the point prediction evaluation: the XGBoost model achieves the lowest (best) SED, followed by the per-stop baseline, and finally the per-category and mean baselines. The results of the SED over the course of a year are consistent with the MAE and RMSE results presented in Figure 5, showing a stable performance advantage for the XGBoost model. Therefore, no figure was included that shows the SED over the course of a year.

5 Discussion

With a Mean Absolute Error (MAE) of approximately 30 seconds, the model's accuracy is competitive with recent studies, such as the 39.9-second MAE reported by Pang et al. [5, p. 363], though it does not reach the sub-10-second error achieved by Li, Daamen, and Goverde [4, p. 18]. It is important to note, however, that direct comparison of performance across different national railway networks is challenging due to inherent differences in operational characteristics and delay distributions. This is in particular true for Li, Daamen, and Goverde [4], as they have only considered one train station [4, p. 11]. Therefore, the primary benchmark for success in this study is the significant improvement over simple, reproducible baselines, a comparison not always present in related work.

The model is quite stable across the time of a year. The notable error drops in the evaluation all have some connection to events that likely caused them. In week 50, the timetable change occurred. In week 8, there is a lot of missing data in the dataset for an unknown reason, but this could well explain the error drop. Finally, in week 20 there was a flood in the south-west in Germany [20]. Week 20 lies in May 2024. Looking at 1, the May 2024 has a big drop in punctuality as well. However, it is unclear, whether the described events completely explain the outliers in the evaluation. Furthermore, if the

extreme weather caused one of the outliers, it would suggest that other extreme weather events in the same year should have produced outliers two, which is not the case.

The general upward trend in the data might suggest the deterioration of the models' performance. But it was not tested, if a more recently trained model would perform any better. It might well be, that the deterioration is mostly an artifact of a changing underlying distribution.

A key limitation identified is the model's difficulty in accurately predicting the magnitude of delay peaks during rush hours, as discussed in Section 4.3. This indicates a limitation in the model's ability to fully represent peak-hour dynamics.

6 Conclusion & Future Work

This paper has presented a stochastic model based on XGBoost for predicting dwell time delay at all German railway stations. The model utilizes a set of relatively simple features derived from timetable data and OpenStreetMap, making it adaptable to other contexts where more specialized data (e.g., track occupation, real-time passenger counts) is unavailable. The model demonstrates solid predictive performance, consistently outperforming several statistical baselines, though the performance is not groundbreaking in any way.

Several avenues for future work should be explored before integrating this model into the proposed chain-of-models framework. First, model accuracy could potentially be improved by incorporating more context-specific features, such as station infrastructure data (e.g., number of platforms) or calendar-based features (e.g., public and school holidays). Second, the current model operates on static timetable data and cannot incorporate real-time information, such as the arrival delay of the train at the station. Since a train's arrival delay directly constrains the possible range of dwell time delay (e.g., a negative dwell time delay is only possible if the train arrives late), incorporating this feature is expected to yield a substantial improvement in predictive accuracy. Lastly, the proposed model's performance should be properly compared to the performance of existing models. A method to exclude the differences in model performance due to different underlying delay distributions should be developed.

References

[1] Deutsche Bahn. *Deutsche Bahn: Pünktlichkeitswerte*. German. Deutsche Bahn. 2025. URL: https://www.deutschebahn.com/de/konzern/konzernprofil/zahlen_fakten/puenktlichkeitswerte-6878476 (visited on 06/10/2025).

[2] Thomas Spanninger et al. "A review of train delay prediction approaches". In: *Journal of Rail Transport Planning & Management* 22 (June 2022), p. 100312. ISSN: 2210-9706. DOI: 10.1016/j.jrtpm.2022.100312.

[3] Pavle Kecman and Rob M. P. Goverde. "Predictive modelling of running and dwell times in railway traffic". In: *Public Transport* 7.3 (June 2015), pp. 295–319. ISSN: 1613-7159. DOI: 10.1007/s12469-015-0106-7.

[4] Dewei Li, Winnie Daamen, and Rob M. P. Goverde. "Estimation of train dwell time at short stops based on track occupation event data: A study at a Dutch railway station". In: *Journal of Advanced Transportation* 50.5 (Apr. 2016), pp. 877–896. ISSN: 2042-3195. DOI: 10.1002/atr.1380.

[5] Zishuai Pang et al. "Dynamic train dwell time forecasting: a hybrid approach to address the influence of passenger flow fluctuations". In: *Railway Engineering Science* 31.4 (June 2023), pp. 351–369. ISSN: 2662-4753. DOI: 10.1007/s40534-023-00311-7.

[6] Theo Döllmann. *Collection of Deutsch Bahn punctuality statistics*. 2025. URL: https://gitlab.com/bahnvorhersage/bahnvorhersage/-/blob/master/data_analysis/db_stats.py (visited on 06/11/2025).

[7] Robert Pardo. *The Evaluation and Optimization of Trading Strategies*. Second. John Wiley & Sons, Ltd, Jan. 2012. ISBN: 9781119196969. DOI: 10.1002/9781119196969.

[8] Hristos Tyralis and Georgia Papacharalampous. "A review of predictive uncertainty estimation with machine learning". In: *Artificial Intelligence Review* 57.4 (Mar. 2024). ISSN: 1573-7462. DOI: 10.1007/s10462-023-10698-8.

[9] Marianna Jacyna, Jolanta Żak, and Piotr Gołębiowski. "The Use of the Queueing Theory for the Analysis of Transport Processes". In: *Logistics and Transport* 41 (2019), p. 101. ISSN: 1734-2015. DOI: 10.26411/83-1734-2015-1-41-12-19.

[10] Marius De Kuthy Meurers and Theo Döllmann. *Jugend Forscht 2020 - TCP: TrAIn_Connection_Precition*. Feb. 2020. URL: https://gitlab.com/bahnvorhersage/docs/-/blob/main/Old%20Docs/JuFo%202020.pdf (visited on 05/29/2025).

[11] Marius De Kuthy Meurers and Theo Döllman. *Jugend Forscht 2021 - TCP: TrAIn_Connection_Prediction*. Mar. 2021. URL: https://gitlab.com/bahnvorhersage/docs/-/blob/main/Old%20Docs/JuFo%202021.pdf (visited on 05/29/2025).

[12] XGBoost developers. *XGBoost documentation: XGBoost Parameters*. URL: https://xgboost.readthedocs.io/en/release_3.0.0/parameter.html (visited on 05/29/2025).

[13] Theo Döllmann. *Jugend Forscht 2023 - Bahn-Vorhersage*. Feb. 2023. URL: https://gitlab.com/bahnvorhersage/docs/-/blob/main/Langfassung_Bahnvorhersage_2023.pdf (visited on 05/29/2025).

[14] Nathaniel Bowditch. *American Practical Navigator*. 2024th ed. Vol. I. Springfield, Virginia: National Geospacial-Intelligence Agency, 2024.

[15] Polipireddy Srinivas and Rahul Katarya. "hyOPTXg: OPTUNA hyper-parameter optimization framework for predicting cardiovascular disease using XGBoost". In: *Biomedical Signal Processing and Control* 73 (Mar. 2022), p. 103456. ISSN: 1746-8094. DOI: 10.1016/j.bspc.2021.103456.

[16] Surjeet Dalal, Edeh Michael Onyema, and Amit Malik. "Hybrid XGBoost model with hyperparameter tuning for prediction of liver disease with better accuracy". In: *World Journal of Gastroenterology* 28.46 (Dec. 2022), pp. 6551–6563. ISSN: 1007-9327. DOI: 10.3748/wjg.v28.i46.6551.

[17] Sayan Putatunda and Kiran Rama. "A Comparative Analysis of Hyperopt as Against Other Approaches for Hyper-Parameter Optimization of XGBoost". In: *Proceedings of the 2018 International Conference on Signal Processing and Machine Learning*. SPML '18. ACM, Nov. 2018, pp. 6–10. DOI: 10.1145/3297067.3297080.

[18] Takuya Akiba et al. "Optuna: A Next-generation Hyperparameter Optimization Framework". In: *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. KDD '19. ACM, July 2019, pp. 2623–2631. DOI: 10.1145/3292500.3330701.

[19] Francis X. Diebold and Minchul Shin. "Assessing point forecast accuracy by stochastic error distance". In: *Econometric Reviews* 36.6–9 (May 2017), pp. 588–598. ISSN: 1532-4168. DOI: 10.1080/07474938.2017.1307247.

[20] Felix Dietzsch. *Regenfluten im Südwesten – ein Zwischenstand*. Deutscher Wetterdienst. May 17, 2024. URL: https://www.dwd.de/DE/wetter/thema_des_tages/2024/5/17.html (visited on 07/10/2025).