EasyChair Preprint

№ 6558

Enhancing Expressiveness of Models for Static Route-Free Estimation of Time of Arrival in Urban Environments

Sören Schleibaum, Jörg P. Müller and Monika Sester

September 7, 2021

ENHANCING EXPRESSIVENESS OF MODELS FOR STATIC ROUTE-FREE ESTIMATION OF TIME OF ARRIVAL IN URBAN ENVIRONMENTS

A PREPRINT

Sören Schleibaum
Department of Informatics
Clausthal University of Technology
soeren.schleibaum@tu-clausthal.de

Jörg P. Müller
Department of Informatics
Clausthal University of Technology
joerg.mueller@tu-clausthal.de

Monika Sester
Institute of Cartography and Geoinformatics
Leibniz University Hanover
monika.sester@ikg.uni-hannover.de

September 6, 2021

ABSTRACT

Scheduling of taxis can reduce cost and potentially decreases CO_2 emissions. However, with a rising number of taxis or travel requests, the time for computing schedules increases. A promising alternative is to estimate trip durations based on historical trip data without calculating routes. Based on an analysis of the state of the art, in this paper we identify and investigate two limitations of route-free Estimated Time of Arrival (ETA) models: First, the overall set of features considered by state of-the-art models is limited. For instance, some potential relevant features (such as weather-related ones) are not considered at all. Also, different models use different sets of features, such as the linear distance between pickup and dropoff location, in diverse and partly inconsistent ways. For those features generally considered, we find different representations, e.g., for trip start time. Second, while discretization of degree-based coordinates for pickup/dropoff locations via spatial binning is very common in state-of-the-art ETA models, the chosen grid cell sizes vary widely and apparently arbitrarily. The contribution of this paper is threefold: First, we propose to enhance route-free ETA models by additional features and investigate the influence of the feature representation on the prediction precision based on a benchmark dataset. Second, we compare different grid cell topologies and sizes as regards their effect on the prediction precision of ETA. Third, we construct and evaluate three types of Machine Learning (ML) models. Our findings indicate that the results outperform state-of-the-art static route-free ETA estimation models.

Keywords Estimated time of arrival · travel time estimation · taxi fleet management · machine learning

1 Introduction

Estimating the travel time for a future taxi trip can be beneficial for scheduling fleets, e.g., in taxi ridesharing. Service providers that manage taxi fleets aim to schedule their taxis optimally to reduce cost and thereby potentially decrease CO_2 emissions. However, for scheduling a fleet, many route alternatives for a set of taxis have to be compared before the actual serving of passengers starts. The underlying computational problems are known to be NP-hard, which is especially relevant for service providers that serve thousands of trips per day. By estimating the time of arrival of future trips, service providers can compare a set of schedules and choose a near-optimal one. That way, a service provider can serve more passengers with fewer taxis, and minimize the total travel time of its taxi fleet.

To enable scheduling, many previous models like (Li et al., 2018) made use of a route or its metadata in ETA. However, other *route-free* models like (Al-Abbasi et al., 2019), (Jindal et al., 2018), (Singh et al., 2019), and (Wang et al., 2019) only consider pickup/dropoff location and additional features like the start time for a single trip. In this paper, we study the route-free instead of route-based ETA due to the following reasons: (1) Route-based ETA is problematic when data is sparse for certain roads (Wang et al., 2019) as the estimation cannot be backed up by historical data. (2) A data-driven approach does not require an underlying road network and is, therefore, easier to realize. (3) Moreover, several taxi trip datasets lack historical trajectories. As we do not depend on the route information, we circumvent the problem of non-existing trajectories and generalize the ETA approach to additional use cases. (4) Once an ML model is trained the usage is relatively cheap, so it forms a less expensive alternative to the more complex calculation of routes (Li et al., 2018). However, the route-free ETA is challenging because the spatial and temporal dependencies of a trip's duration still depend on a given urban environment. Hence, influences like daily travel patterns must be learned indirectly, which makes the problem hard. As we focus on the *static* ETA (information is limited to those available before the trip) the problem is even more complicated.

Remarkably, the majority of the considered ETA models use grid indices as a replacement of exact degree-based coordinates for pickup/dropoff locations. From our perspective, using a grid does necessarily improve the prediction precision in ETA models because some of the location information present in the degree-based coordinates is unused. As regards additional features, we observe that several variants for the start time representation are used but not compared; while some other features like the Euclidean distance of a trip are included by only one model, others like weather-related ones are ignored.

This works aims to investigate the influence of additional features and the representation of pickup/dropoff locations on the prediction precision of static route-free ETA in urban environments; Therefore, we collect potential features and their representations from related work and enhance a historic taxi trip dataset by them. Next, we compare the influence of the features and their representations on the prediction precision of multiple ML methods commonly used for ETA. Based on that, we select an optimal set of features for each of the methods. Specifically, three main research questions are addressed: (RQ1) *Does enhancing existing taxi trip datasets by additional features like variants of start time and distance or weather-related ones improve the prediction precision?* (RQ2) *Should the degree-based coordinates be used directly, or should a grid-based proxy be used?* (RQ3) *How does the topology (square, triangle, or hexagon) and size of a grid cell influence the prediction precision?*

2 Related Work

Route-free Models As part of a ridesharing framework, Al-Abbasi et al. (2019) develop a route-free ETA model that receives the start time, passenger count, and indices of a 150-meter square grid for pickup/dropoff location as input. The model is a relatively simple two-layer Fully-Connected Feedforward Neural Network (FCNN) with 64 neurons in each layer. de Araujo and Etemad (2019) also build a similar FCNN with a different architecture; their FCNN consists of five layers with 500, 200, 50, 50, and 50 corresponding neurons. The input features are the degree-based coordinates for pickup/dropoff, the hour and weekday encoded in sine/cosine to capture the circular nature of these features, the distance between pickup and dropoff, and an estimated travel distance with a map-based locally-linear regressor. As part of a transportation service, Jindal et al. (2018) develop an FCNN model that can estimate the trip distance as well as trip duration; for the ETA, the last layer of the trip distance estimator is enhanced by a 10-minute time-bin with Weekday Separation (WDS). Similar to using grids, temporal binning separates a day into several bins. For instance, when a 10-minute time-bin is considered, a day is divided into 10-minute time-bins, resulting in 144 time-bins. For time-bins with WDS, weekend time-bins are added on top of the ones for workdays. Therefore, the information about a trip starting on a workday or at the weekend is inherently captured. As representation for pickup/dropoff location, Jindal et al. (2018) use a 200-meter square grid. Based on (Al-Abbasi et al., 2019), Singh et al. (2019) exchange the FCNN by a Random Forest (RF) to perform the ETA in a ridesharing framework. Singh et al. (2019) use the weekday and start time of a trip together with the degree-based coordinates as input features. Wang et al. (2019) develop a neighbor-based method that estimates the duration of a trip by considering trips with similar pickup/dropoff locations and start time. They use the start time, Euclidean distance, and the indices of a 50-meter square grid as input features.

Route-based Models Kankanamge et al. (2019) apply XGBoost to develop a route-based ETA model. The indices of a 100-meter square grid for pickup/dropoff, a 10-minute time-bin, the driver and medallion identifier, and multiple features that are derived from the route like the number of intersections are the input features. Another model proposed by Li et al. (2018) inputs a 10-minute time-bin, the indices of square grid for pickup/dropoff, and multiple information extracted from the route into their multi-task representation learning model.

Table 1: Overview of properties of existing ETA models

Reference	Method	Route-based	Additional Features	Grid topology	Grid cell size
Al-Abbasi et al. (2019)	FCNN	✗	Start time, passenger count	Square	150 meters
de Araujo and Etemad (2019)	FCNN	✗	Sine/cosine of weekday and hour, distance, estimated distance	-	-
Jindal et al. (2018)	FCNN	✗	10-min. time-bin with WDS, estimated distance	Square	200 meters
Kankanamge et al. (2019)	XGBoost	(✓)	10-min. time-bin, driver identifier, medallion identifier	Square	100 meters
Li et al. (2018)	Representation learning	(✓)	10-min. time-bin	Square	-
Singh et al. (2019)	RF	✗	Weekday, time of day	-	-
Wang et al. (2019)	Neighbor-based	✗	Start time, Euclidean distance	Square	50 meters

Research Gap An overview of the described ETA models is given in Table 1. Remarkably, except for (de Araujo and Etemad, 2019; Singh et al., 2019), all mentioned models use grid indices as a replacement of the degree-based coordinates for the pickup/dropoff locations. While these models achieve impressive prediction precision, they lack a convincing argumentation for favoring a grid with a certain cell size or topology over degree-based coordinates. While using a grid can be assumed to be useful for tasks like demand prediction, it is unclear whether it is so for the case of ETA – and what a good topology or resolution would be in this case. As regards additional features, we observed that e.g. for the trip start time, several representations – like time-bins or common date/time representation – were used but not compared. Some features were only included sporadically, like the euclidean distance by Wang et al. (2019). Other potentially relevant features, like different time-bin variants, other distance measures such as Manhattan distance, and weather-related features like precipitation (influences vehicle speed [Feng et al., 2020]) are ignored completely.

3 Methodology

3.1 Taxi Trip Dataset

Dataset Description The NYC taxi trip dataset may well be the largest publicly available dataset about taxi trips. Currently, trips between 2009 and 2020 can be downloaded at City of New York (2020). The number of trips is large; alone from January 2016 to June 2018, about 780 million trips were recorded with more than a million trips per day at the end of 2018 (de Blasio and Joshi, 2018). The exact locations are only available for older trips. Therefore, in this paper, we use the Yellow taxi trip subset from January till May 2016 for training/validation and the same months from 2015 for testing. Overall, we select 3.7M trips for training/validation and 250K for testing randomly.

Outlier Removal In the taxi trip dataset several erroneous or unrealistic trips are included. Therefore, we define multiple criteria to remove outliers. For each criterion, we report the percentage of records affected; overall, 3.26% of the trips are excluded by the criteria: (1) *Location is outside the study area* - 1.78% – Because our approach for ETA is data-driven, we only handle those areas in which trips were recorded previously. Therefore, we exclude trips that are not in a square with the bottom left at 40.587917, -74.089829 and top right at 40.901386, -73.68566. (2) *Location is not in a district* - 1.84% – Some pickup or dropoff points of trips are not in a community district. With this criterion, we are, for instance, able to exclude points that were mistakenly recorded on areas of water like the Hudson River. (3) *Duration is unreasonably high* - 0.13% – We consider approximately three hours as the longest possible trip duration because this represents covering the longest possible distance in the grid with the average taxi travel speed for Manhattan central business district. Therefore, we filter all trips that exceed three hours. (4) *Duration is zero or less than 30s* - 0.51% – Moreover, we exclude trips with a duration of zero or close to zero seconds. (5) *Distance is zero* - 1.52% – We remove trips in which the distance between their pickup and dropoff point is zero. (6) *Duration and distance do not correlate reasonably* - 0.7% – This inconsistency refers to two cases. First, the duration is much higher than the traveled distance. This might be the case when some sightseeing is done. Second, the duration can be relatively low despite a large distance.

Additional Features As described previously, different models for ETA consider diverse sets of input features. In general, we orientate on the additional features from related work - listed in Table 1 - and enhance this set by alternative representations of the same features and additional features like precipitation that potentially influence the prediction precision. While we expect that not all features or their representations will increase the prediction precision, we will back up such intuition in later experiments. Because we perform static ETA, we can enhance the New York City (NYC) taxi trip dataset only by features that are known before the trip starts. Therefore, we exclude features like the number of traffic lights on a route or the price, which indirectly capture the length of the route. To investigate RQ1, we consider the *year*, *month*, *week*, *day*, *weekday*, *hour*, and *minute* of a trip as time-based features. Also, we compare different time-bins with sizes from 5 to 55 minutes with and without WDS. Similar to Wang et al. (2019), we also include

Table 2: Overview of categorized additional features

Category	Features
Basic	*Coordinates*, and *grid* indices for pickup/dropoff
Time-based	*Year*, *month*, *week*, *day*, *weekday*, *hour*, *minute*, and variants for *time-bins*
Distance-based	*Haversine* and *Manhattan* distance
Weather-based	*Barometer*, *cloudy*, *humidity*, *precipitation*, *snow*, *temperature*, *visibility*, and *wind*
Other	*Community district*, *passenger count*

distance-based features like the Manhattan distance. However, unlike Wang et al. (2019) we do not use the euclidean distance between pickup and dropoff location, but rather favor the Haversine distance which captures the spherical shape of the earth. As described by Feng et al. (2020), weather-based features can influence the speed of vehicles. Consequently, we take into account weather-based features at the hour a trip starts. We consider *barometer*, *cloudy*, *humidity*, *precipitation*, *snow*, *temperature*, *visibility*, and *wind*. Moreover, for instance, Wang et al. (2019) showed that different regions have dissimilar average speeds. Based on that, we assume an influence of the region on the travel time and study the influence of using the one-hot encoded identifier of a *community district* in which a trip starts/ends. A feature used by Al-Abbasi et al. (2019) and also considered by us is the *passenger count*. In contrast to Kankanamge et al. (2019), we are not able to include the *driver* or *medallion identification* because these features are not available in all of the considered subsets of the NYC taxi trip dataset. To evaluate the influence of the grid topology and cell size (RQ2/3), we consider the degree-based coordinates and square, triangle, and hexagon topology with cell sizes from 5 to 1000 meters for pickup/dropoff location. An overview of the categorized features is outlined in Table 2.

3.2 Experiments

To tackle the raised research questions, we run four experiments: (EXP1) To determine potential additional features (RQ1), we add every feature to the degree-based coordinates and measure the prediction precision. (EXP2) To evaluate the influence of the grid topology and cell size (RQ2/3), we compare the prediction precision achieved with degree-based coordinates to the one accomplished with each of the grid variants. (EXP3) After EXP1 several alternatives for time- and distance-based features are reasonable. Therefore, we compare those in combination with the selected grid and choose a final feature set (RQ1). (EXP4) Next, the hyperparameters of all ML methods are tuned and the achieved prediction precision is reported. In the following, we describe these experiments in more detail. Before that, we outline the chosen ML methods. In the end, we also list the metrics for evaluating the prediction precision.

ML Methods The main objective of this paper is not to develop new methods but rather to study the influence of different features and their representation on state-of-the-art ML methods. Therefore, we chose three sophisticated ML methods: the RF, XGBoost, and FCNN. The RF uses *bagging* or the learning of multiple models or decision trees and the combination of these models. The XGBoost, on the other hand, is based on *boosting*, which means the iterative building of models based on the errors of the previous ones. The third method, an FCNN, is a popular technique particularly successful when trained based on large datasets. As shown in Table 1, all three ML methods were used previously to tackle the ETA problem.

EXP1 - Additional Features To determine additional features that can increase the prediction precision, we add every non-basic feature solely to the degree-based coordinates (baseline) of the taxi trip dataset. Next, we train ten RF, XGBoost, and FCNN models for each feature subset to get a more representative prediction precision and compare this result to the baseline. For the RF and XGBoost, we do not set any hyperparameters because these methods usually achieve good results without tuning them; the number of trees for both ML methods is set to 100. For the FCNN models, we use a relatively simple fully-connected feedforward architecture with three layers (128, 64, and 32 corresponding neurons) and a rectifier linear unit as the activation function for all layers. We set the batch size to 10K and the number of epochs to 25 to limit the training time. The learning rate is set to 0.1.

We use 500K trips from January to May for training (400K) and validation (100K). To verify the increase of the prediction precision, we apply a two-sided t-test and compare its p-values p to alpha α, which we set to 0.001 to cover all common significance levels. Finally, we correct p with the Bonferroni method (see [Armstrong, 2014]) to avoid the multiple comparison problem. If adding a feature reduces one of the evaluation metrics significantly, we consider it for EXP3 and EXP4.

EXP2 - Degree-Based Coordinates vs. Grid Indices To study the influence of the representation of the pickup/dropoff location on the prediction precision, we train models of the three ML methods for several grid

variants without including any other features and compare the results to the baseline which consists of the degree-based coordinates. As grid variants, we consider square, triangle, and hexagon topology with grid cell sizes from 5 to 1000 meters. We also analyze the relative change of the training time to better understand the motivations of others that use grid indices for spatial data. Otherwise, the experimental setup is similar to the one from EXP1.

EXP3 - Combination of Additional Features with Coordinate-Representation After EXP1 several alternative representations achieve similar results. To choose the most promising alternative for the EXP4, we build ML models and compare the alternative representations. Similar as before, we build ten models each, verify changes in the prediction precision with a two-sided t-test, and correct the resulting p-values. In contrast to EXP1 and EXP2, we increase the number of trips to 1M and the number of trees for the RF and XGBoost to 200. For the FCNN we insert an additional first layer with 256 neurons.

EXP4 - Hyperparameter Tuning and Larger Scale Based on the results from EXP3, we tune the hyperparameters with Bayesian optimization; as acquisition function we use expected improvement. We train models on a larger scale by increasing the size of the training dataset from 0.8M to 1.2M trips. For the RF, as recommended by Probst et al. (2019), we tune the minimum number of samples for splitting a node and for leaf nodes. Additionally, we tune the maximum number of features per split and the maximum depth of a tree. The number of trees in the forest is set to 300 to get a reasonable performance gain. Similar to Probst et al. (2019) we do not consider this parameter as tunable because, in general, a larger number of trees is always desirable. For hyperparameter-tuning of the XGBoost, we orientate on Wang and Ni (2019) who tune among others the following hyperparameters: (1) the maximum depth of a tree, (2) the minimum weights of instances needed in a child, (3) the subsample ratio of the training data per tree, (4) the minimum loss reduction required for making a further partition on a child, and (5) the subsample ratio of features when constructing a tree. We apply a similar hyperparameter-tuning strategy and set the number of trees also to 300. To tune the hyperparameters of the FCNN, we consider the number of hidden layers, the number of neurons included in those, the learning rate of the used Adam optimization algorithm, and the batch size. The final FCNN models are trained for 50 epochs. Once the ML models are trained, we evaluate them on previously unused test data from mid-May to June 2015 and report their results.

Evaluation metrics We select those evaluation metrics that are common for regression tasks and used by other models to enable comparing their results to ours. Using the Mean Absolute Error ($MAE = {}^1/_N \sum_i |y_i - \hat{y}_i|$) and the Mean Relative Error ($MRE = \sum_i |y_i - \hat{y}_i| / \sum_i y_i$) is reasonable because multiple models evaluate their performance based on at least one of them. Similar to Li et al. (2018) we will also use the Mean Absolute Percentage Error ($MAPE = {}^1/_N \sum_i |(y_i - \hat{y}_i)/y_i|$), which is robust to outliers and its values are easy to understand.

4 Results

When not stated otherwise, the reported results for the four experiments are verified via a two-sided t-test or significant. For EXP1 and EXP2, we present only the relative changes of the prediction precision and training time rather than their concrete values because the models are otherwise hard to compare due to their small size.

EXP1 - Additional Features In Table 3, we visualize the relative change for the MAE, MRE, and MAPE together with the Bonferroni corrected p-values for the considered features compared to the baseline that consists of the degree-based coordinates. Negative values indicate an improvement of the prediction precision or a decrease of an evaluation metric. Features that reduce one of the evaluation metrics significantly are marked with a ρ for the RF, a χ for the XGBoost, and a ν for the FCNN after the feature name. To limit the size of the table, we only visualize the two most promising time-bin (TB) variants with and without weekday separation (WDS).

For the RF, the *month* decreases the MRE significantly (indicated by three asterisks) by 0.81 percentage points. Other features that decrease one or multiple evaluation metrics for the RF are the *week*, *weekday*, *hour*, and all time-bin variants. Interestingly, including a distance measure decreases the prediction precision. As regards the remaining features, only the *temperature* increases the prediction precision. For the XGBoost, *weekday*, *hour*, the time-bin variants, and the distance measures increase the prediction precision. For the FCNN, only the distance measures increase the prediction precision. For the RF and XGBoost, we consider solely the features that increase their prediction precision in EXP3 and EXP4. For the FCNN, we additionally take all features into account that decreased at least one evaluation metric for one method because learning the trip duration with the relatively small dataset in EXP1 is more challenging for the FCNN.

EXP2 - Degree-Based Coordinates vs. Grids Indices In Figure 1, we visualize the relative change of prediction precision measured by the MAPE and training time for square grids with different grid cell sizes or heights compared to

Table 3: EXP1 - Relative change of the prediction precision compared to the baseline of the degree-based coordinates; negative values show an improvement of the prediction precision in percentage points. The results are verified via t-test with Bonferroni-corrected p-values.

	RF			XGBoost			FCNN		
Feature	ΔMAE	ΔMRE	ΔMAPE	ΔMAE	ΔMRE	ΔMAPE	ΔMAE	ΔMRE	ΔMAPE
Year	0.0^{ns}	0.04^{ns}	0.06^{ns}	40.03^{ns}	0.0^{ns}	0.17^{ns}	0.23^{ns}	0.19^{ns}	1.95^{ns}
Month $^{\rho}$	-0.8^{*}	-0.81^{***}	-0.3^{ns}	40.24^{ns}	0.0^{ns}	0.46^{ns}	0.31^{ns}	0.19^{ns}	-1.0^{ns}
Week $^{\rho}$	-1.09^{***}	-0.99^{***}	0.03^{ns}	-0.58^{ns}	-0.71^{**}	0.06^{ns}	0.53^{ns}	0.35^{ns}	2.49^{ns}
Day	-0.24^{ns}	-0.29^{ns}	0.72^{**}	0.4^{ns}	0.32^{ns}	0.87^{***}	0.21^{ns}	0.15^{ns}	2.09^{ns}
Weekday $^{\rho\chi}$	-3.38^{***}	-3.41^{***}	-1.92^{***}	-2.82^{***}	-2.82^{***}	-2.48^{***}	0.43^{ns}	0.17^{ns}	-0.78^{ns}
Hour $^{\rho\chi}$	-13.33^{***}	-13.33^{***}	-10.39^{***}	-12.84^{***}	-12.84^{***}	-11.66^{***}	0.2^{ns}	0.08^{ns}	1.49^{ns}
Minute	-0.21^{ns}	-0.11^{ns}	0.69^{**}	0.61^{ns}	0.46^{ns}	0.92^{*}	0.23^{ns}	0.12^{ns}	1.76^{ns}
TB 5 min. $^{\rho\chi}$	-14.01^{***}	-13.91^{***}	-10.39^{***}	-13.13^{***}	-13.23^{***}	-11.69^{***}	0.12^{ns}	0.08^{ns}	3.48^{ns}
TB 40 min. $^{\rho\chi}$	-14.02^{***}	-13.88^{***}	-10.84^{***}	-12.98^{***}	-13.05^{***}	-11.61^{***}	0.4^{ns}	0.17^{ns}	1.58^{ns}
TB 5 min. WDS $^{\rho\chi}$	-16.55^{***}	-16.55^{***}	-13.03^{***}	-16.08^{***}	-16.19^{***}	-14.46^{***}	0.03^{ns}	0.0^{ns}	2.16^{ns}
TB 15 min. WDS $^{\rho\chi}$	-16.57^{***}	-16.4^{***}	-12.82^{***}	-16.08^{***}	-16.01^{***}	-14.58^{***}	0.2^{ns}	-0.06^{ns}	0.45^{ns}
Haversine $^{\chi\nu}$	0.22^{ns}	0.33^{ns}	0.81^{***}	0.31^{ns}	0.25^{ns}	-1.21^{***}	-40.12^{***}	-40.13^{***}	-46.74^{***}
Manhattan $^{\chi\nu}$	0.35^{ns}	0.37^{ns}	1.05^{***}	0.38^{ns}	0.32^{ns}	-0.92^{***}	-38.68^{***}	-38.73^{***}	-48.38^{***}
Barometer	-0.26^{ns}	-0.26^{ns}	0.81^{**}	0.47^{ns}	0.43^{ns}	0.95^{***}	0.36^{ns}	0.21^{ns}	4.15^{*}
Cloudy	-0.09^{ns}	-0.11^{ns}	0.15^{ns}	0.17^{ns}	0.11^{ns}	0.32^{ns}	0.08^{ns}	0.13^{ns}	0.71^{ns}
Humidity	-0.32^{ns}	-0.26^{ns}	0.78^{***}	0.67^{ns}	0.46^{ns}	1.24^{***}	0.43^{ns}	0.25^{ns}	2.01^{ns}
Precipitation	-0.18^{ns}	-0.07^{ns}	-0.06^{ns}	0.21^{ns}	0.18^{ns}	0.4^{ns}	0.4^{ns}	0.29^{ns}	-1.19^{ns}
Snow	-0.19^{ns}	-0.18^{ns}	-0.36^{ns}	0.02^{ns}	-0.04^{ns}	0.14^{ns}	0.22^{ns}	0.06^{ns}	2.36^{ns}
Temperature $^{\rho}$	-1.16^{***}	-1.06^{***}	0.21^{ns}	0.1^{ns}	0.0^{ns}	0.61^{ns}	0.39^{ns}	0.25^{ns}	2.66^{ns}
View	-0.13^{ns}	-0.04^{ns}	0.06^{ns}	0.1^{ns}	0.04^{ns}	0.23^{ns}	0.06^{ns}	0.12^{ns}	1.83^{ns}
Wind	-0.15^{ns}	-0.11^{ns}	0.63^{*}	0.32^{ns}	0.39^{ns}	0.72^{**}	0.22^{ns}	0.19^{ns}	2.84^{ns}
Community dist. $^{\nu}$	-0.05^{ns}	-0.07^{ns}	0.03^{ns}	-0.17^{ns}	-0.21^{ns}	-0.29^{ns}	0.9^{ns}	3.7^{ns}	1.96^{ns}
Passenger count	0.03^{ns}	0.04^{ns}	0.39^{ns}	0.25^{ns}	0.21^{ns}	0.49^{ns}	0.23^{ns}	0.19^{ns}	3.37^{ns}

$^{***}p \leq 0.001$, $^{**}p \leq 0.01$, $^{*}p \leq 0.05$, $^{ns}p > 0.05$;
features that decrease an evaluation metric significantly are marked with $^{\rho}$ for RF, $^{\chi}$ for XGBoost, and $^{\nu}$ for FCNN

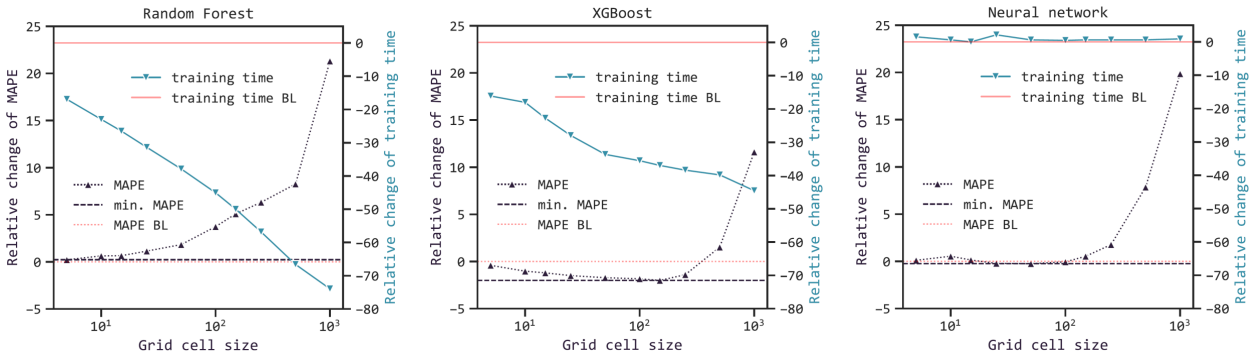

Figure 1: EXP2 - Influence of the grid cell size on the MAPE and training time with comparison to the degree-based coordinates or baseline (BL)

the baseline of degree-based coordinates. The results from all three ML are shown but other evaluation metrics and grid topologies are omitted because their results are very similar. For the RF, as shown on the leftmost graphic, we observe that with grid cell sizes larger than 15 meters, the prediction precision decreases compared to the baseline. Interestingly, at the same time, using the grid indices instead of the degree-based coordinates reduces the training time. For the XGBoost, we notice a similar behavior regarding the training time; however, the prediction precision is significantly better when using grid cell sizes between 15 and 250 meters compared to the baseline and smaller/larger grid cell sizes. For the FCNN, we observe a similar pattern in the prediction precision as for the RF. However, the largest possible grid cell size that does not decrease the prediction precision is 250 meters. As expected, different grid cell sizes do not change the training time for the FCNN.

EXP3 - Combination of Additional Features with Coordinate-Representation When we compare the alternative start time representations gathered in EXP1 for the RF, the time-bin features with the WDS decrease the prediction precision; we favor the *5-minute time-bin* over the *40-minute time-bin* and the combination of *hour* and *minute* because it decreases all evaluation metrics. For the XGBoost, we choose the *5-minute time-bin*, which achieves the best prediction precision, even though the difference to the combination of *hour* and *minute* is not significant. Moreover, we choose the Haversine distance instead of the Manhattan distance because it achieves a higher prediction precision. For the FCNN, the time-bin variants with the WDS decrease the prediction precision. While the difference between the combination of

Table 4: Comparison of our ML models to others based on different evaluation metrics - we trained ten models each and report the mean values

	Evaluation metric	MAE	MRE	MAPE
Our models	RF	182.82	22.2532	28.374
	XGBoost	182.0264	22.1566	26.8838
	FCNN	**175.3038**	**21.3513**	**23.119**
Other rebuild models	Al-Abbasi et al. (2019)	199.4434	24.2913	27.4652
	de Araujo and Etemad (2019)*	201.5998	24.554	28.1508
	Jindal et al. (2018)*†	185.9265	22.5607	23.8429
	Singh et al. (2019)	185.3999	22.5673	28.3598

* real trip distance instead of their estimated trip distance; † grid cell size of 150 meters instead of 200 meters

hour and *minute* is not significant, on average the *5-minute time-bin* slightly increases the prediction precision and is therefore chosen. Based on the same argument, we select the *Haversine* over the *Manhattan* distance.

Based on the results of EXP1, EXP2, and EXP3, we choose different feature sets for the ML methods. The final feature set for the RF consist of the degree-based coordinates, *month*, *week*, *weekday*, a *5-minute time-bin*, and the *temperature*. For the XGBoost, we use the *weekday*, a *5-minute time-bin*, the *Haversine* distance, and the indices of a 50-meter square grid. For the FCNN, the final feature set consists of the *month*, *week*, *weekday*, a *5-minute time-bin*, the *Haversine* distance, and the degree-based coordinates.

EXP4 - Hyperparameter Tuning and Larger Scale After hyperparameter tuning for the RF, we set the minimum number of samples for splitting a node to four and the one for leaf nodes to four; the maximum number of features per split is chosen automatically and the maximum depth of a tree is 89. For the XGBoost, we set the maximum depth of a tree to eleven, the minimum weights of instances needed in a child to seven, the subsample ratio of the training data per tree to one, the minimum loss reduction required for making a further partition on a child to zero, and the subsample ratio of features for a tree to one. As regards the FCNN, we use a four-layer FCNN with 300, 150, 50, and 25 corresponding neurons. The batch size is set to 128 and the learning rate to 0.001. As shown in Table 4, the FCNN outperforms the other ML models in all evaluation metrics with an MAE of 175s, an MRE of 21 percent, and a MAPE of 23 percent.

5 Discussion

RQ1 - Additional Features Our results show that including additional features can increase the prediction precision. We compared several representations for the trip start time, which has in general and as expected a relatively large influence on the prediction precision. Interestingly, using time-bins with WDS as Jindal et al. (2018) proposed did not increase the prediction precision for any ML method; time-bin without WDS slightly increased the prediction precision for all ML methods. Using a distance measure does improve the prediction precision for the XGBoost and FCNN; remarkably, for the RF doing the same decreases the prediction precision slightly. An investigation of the feature importances for the RF models with and without the distance shows that those with distance mainly estimate a trip duration based on it. As a result, the other information about pickup/dropoff location only has a small influence and the prediction precision decreases due to the decreased information. We tried many different weather-based features with an hourly resolution; surprisingly, including the *precipitation*, which influences the traffic speed (Feng et al., 2020), does not increase the prediction precision for ETA. Only *temperature* increased the prediction precision. This could be caused by a correlation between the *temperature* and traffic density which influences the trip duration. While the *community district* no positive influence on the prediction precision, we want to point out that such a district-based feature can be particularly useful for removing outliers with unrealistic pickup/dropoff locations like in a river. Overall, we observed that many models used different sets of features and lacked sufficient argumentation for the included/excluded features. Based on our findings, we recommend using our analysis as a basis for such argumentation. While the concrete feature representation has only a slight effect, including/excluding a feature, like distance measures, depends on the ML method chosen.

RQ2/3 - Degree-based Coordinates vs. Grid-Based Proxy Grids with a relatively small grid cell size show similar prediction precision to the degree-based coordinates. The largest grid cell size that performs equally well with our relatively small models is 15 meters for the RF and 250 meters for the FCNN. Interestingly, for the XGBoost a grid cell size of 50 meters performs better than the degree-based coordinates. We expect that these numbers are lower when the

model size or the training time is increased. Larger grid cell sizes decrease the prediction precision. This pattern was also observed for the hexagon and triangle grid topology.

Interestingly, when no or not all hyperparameters are fixed and the method chooses them independently, like we did with RF and XGBoost, increasing the grid cell size can lead to lower training time. While the decreased training time might be beneficial for some, the inference time of ML models matters more in practice. The inference time, however, does not depend on the grid cell size but the architecture of the ML model. Moreover, converting the degree-based coordinates into grid indices requires a transformation step. Except for (de Araujo and Etemad, 2019; Singh et al., 2019), the related works considered in this paper perform such transformation and favor grid indices over the degree-based coordinates. Building on our results, we recommend omitting this transformation for the RF and FCNN. Instead, training an ETA model directly on the degree-based coordinates achieves equally good or increases the prediction precision.

Comparison to Other Models An overview of the comparison between our ETA models and the ones of others rebuild by us is outlined in Table 4. When comparing our results to others, the rebuild of the model from Al-Abbasi et al. (2019) achieves an MAE of 199s on our data. Therefore, the model is on average around 24s less precise per trip than our FCNN model or around 17s when compared to the RF and XGBoost. We observe a very similar result with the model proposed by de Araujo and Etemad (2019). We did not rebuild their distance estimator due to time constraints but used the actual trip distance directly as an input. Since the actual trip distance is the input for their estimator, we do not expect different results when rebuilding their approach completely. Based on the same argument we also use the actual trip distance for the FCNN-based model proposed by Jindal et al. (2018). As regards the MAE and MRE, their model is slightly less accurate compared to our RF and XGBoost ones; however, their MAPE is lower. Nevertheless, our FCNN still performs better in all evaluation metrics. The RF model from Singh et al. (2019) performs not as good as our FCNN but is close compared to our other models. This result is remarkable given the effort we put into carefully selecting the input features. As regards our RF, we believe that the small increase in prediction precision of 2.6s for MAE when using a 5-minute time-bin instead of the combination of hour and minute is compromised by lower temperatures in the test data from 2015 compared to the training data from 2016 that are unknown by the RF. Wang et al. (2019) report an MAE of 196s for their neighbor-based model on the same test data; our MAE is 8 to 21 seconds lower. Even though the results for the RF are similar to the ones from Singh et al. (2019), we can outperform all considered static route-free approaches for ETA with our FCNN.

Limitations and Future Work This work discusses five categories representing around 35 features or feature representations and additional 30 grid variants. However, other features like accident data might also be beneficial. Besides that, many more feature representations are possible; for instance, time-based features like the *hour* can be converted via a sine/cosine function to make the difference between 11:00 p.m. and 00:00 a.m similar to the distance between any other neighboring hours as done by de Araujo and Etemad (2019). Moreover, potential features can be combined in different ways, which might lead to a better model. Therefore, we did not provide a complete overview of potential features, their representation, or combination, but rather a good orientation for future research. Further, we used only data from Yellow taxi cabs in New York City. Even though we expect similar results with other datasets, especially the prediction precision will differ.

We argued that using an ML model is a less expensive alternative to calculating the real route. While this might be true in theory, we assume that for many smaller scenarios in which trajectories are available, route-based models are a reasonable alternative. A good starting point might be the work of Yang et al. (2018). As already argued by Kankanamge et al. (2019), the travel time also depends on the taxi and driver. In the dataset used for training, there was no information about the driver/taxi available. However, we believe that individualizing ETA might be a promising direction to further increase the prediction precision. While achieving a high prediction precision is desirable, other objectives might also be relevant. For instance, we might consider a taxi ridesharing system in which the service provider uses our best ETA model to plan the schedules of taxis. Previously, we argued in Schleibaum and Müller (2020) an increase in the usage of a ridesharing system could be achieved by increasing user satisfaction. One option proposed is to explain decisions of the system by providing insights into its components like ETA. Therefore, future work could consider the explainability of ETA models.

6 Conclusion

Scheduling is an important part of taxi fleet management that can reduce cost and CO_2 emissions. To improve scheduling, in this paper, we presented several ML models that estimate the duration of a trip in a static route-free manner in a given urban environment. Moreover, we showed the potential of additional features like the temperature and choosing the best representation of the start time. Furthermore, we compared different grid variants and conclude that a relatively small grid cell size can be used. However, using the degree-based coordinates directly achieves the

same prediction precision for the RF and FCNN and saves a data transformation step. Afterwards, we argued that other models chose a sub-optimal set of features and their representation. We backed this up by performing better than previous static route-free ETA models with an FCNN.

Acknowledgments

This work was supported by the Deutsche Forschungsgemeinschaft under grant 227198829/GRK1931. The SocialCars Research Training Group focuses on future mobility concepts through cooperative approaches.

References

Al-Abbasi, A. O., Ghosh, A., Aggarwal, V., 2019. DeepPool: Distributed Model-Free Algorithm for Ride-Sharing Using Deep Reinforcement Learning. IEEE Transactions on Intelligent Transportation Systems 20, 4714–4727. https://doi.org/10.1109/TITS.2019.2931830

Armstrong, R.A., 2014. When to use the Bonferroni correction. Ophthalmic Physiol Opt 34, 502–508. https://doi.org/10.1111/opo.12131

City of New York, 2020. TLC Trip Record Data.

de Araujo, A. C., Etemad, A., 2019. Deep Neural Networks for Predicting Vehicle Travel Times, in: 2019 IEEE SENSORS. Presented at the 2019 IEEE SENSORS, IEEE, Montreal, QC, Canada, pp. 1–4. https://doi.org/10.1109/SENSORS43011.2019.8956878

de Blasio, B., Joshi, M., 2018. 2018 Factbook.

Feng, Y., Brenner, C., Sester, M., 2020. Learning a Precipitation Indicator from Traffic Speed Variation Patterns. Transportation Research Procedia 47, 203–210. https://doi.org/10.1016/j.trpro.2020.03.090

Jindal, I., Qin, Z. T., Chen, X., Nokleby, M., Ye, J., 2018. Optimizing Taxi Carpool Policies via Reinforcement Learning and Spatio-Temporal Mining, in: 2018 IEEE International Conference on Big Data (Big Data). IEEE, Seattle, WA, USA, pp. 1417–1426. https://doi.org/10.1109/BigData.2018.8622481

Kankanamge, K. D., Witharanage, Y.R., Withanage, C. S., Hansini, M., Lakmal, D., Thayasivam, U., 2019. Taxi Trip Travel Time Prediction with Isolated XGBoost Regression, in: 2019 Moratuwa Engineering Research Conference (MERCon). Presented at the 2019 Moratuwa Engineering Research Conference (MERCon), IEEE, Moratuwa, Sri Lanka, pp. 54–59. https://doi.org/10.1109/MERCon.2019.8818915

Li, Y., Fu, K., Wang, Z., Shahabi, C., Ye, J., Liu, Y., 2018. Multi-task Representation Learning for Travel Time Estimation, in: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining. ACM, London United Kingdom, pp. 1695–1704. https://doi.org/10.1145/3219819.3220033

Probst, P., Wright, M. N., Boulesteix, A.-L., 2019. Hyperparameters and tuning strategies for random forest. Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 9. https://doi.org/10.1002/widm.1301

Schleibaum, S., Müller, J. P., 2020. Human-Centric Ridesharing on Large Scale by Explaining AI-Generated Assignments. Presented at the GoodTechs '20, ACM, Antwerp, Belgium. https://doi.org/10.1145/3411170.3411274

Singh, A., Al-Abbasi, A. O., Aggarwal, V., 2019. A Reinforcement Learning Based Algorithm for Multi-hop Ridesharing: Model-free Approach. Presented at the NeurIPS 2019 Workshop, Vancouver, BC, Canada.

Wang, H., Tang, X., Kuo, Y.-H., Kifer, D., Li, Z., 2019. A Simple Baseline for Travel Time Estimation using Large-Scale Trip Data. ACM Transactions on Intelligent Systems and Technology 10, 1–22. https://doi.org/10.1145/3293317

Wang, Y., Sherry Ni, X., 2019. A XGBoost Risk Model via Feature Selection and Bayesian Hyper-Parameter Optimization. IJDMS 11, 01–17. https://doi.org/10.5121/ijdms.2019.11101

Yang, B., Dai, J., Guo, C., Jensen, C. S., Hu, J., 2018. PACE: a PAth-CEntric paradigm for stochastic path finding. The VLDB Journal 27, 153–178. https://doi.org/10.1007/s00778-017-0491-4