Hindawi
Journal of Advanced Transportation
Volume 2024, Article ID 9301691, 14 pages
https://doi.org/10.1155/2024/9301691

Research Article

An Explainable Stacked Ensemble Model for Static Route-Free Estimation of Time of Arrival

Sören Schleibaum,[1] **Jörg P. Müller**,[1] **and Monika Sester**[2]

[1]*Clausthal University of Technology, Clausthal-Zellerfeld, Germany*
[2]*Leibniz University Hannover, Hanover, Germany*

Correspondence should be addressed to Sören Schleibaum; soeren.schleibaum@tu-clausthal.de

Received 5 September 2022; Revised 24 December 2023; Accepted 28 December 2023; Published 7 March 2024

Academic Editor: Domokos Esztergár-Kiss

Sustainable concepts for on-demand transportation, such as ridesharing or ridehailing, require advanced technologies and novel dynamic planning and prediction methods. In this paper, we consider the prediction of taxi trip durations, focusing on the problem of the estimated time of arrival (ETA). ETA can be used to compute and compare alternative taxi schedules and to provide information to drivers and passengers. To solve the underlying hard computational problem with high precision, machine learning (ML) models for ETA are the state of the art. However, these models are mostly *black box* neural networks. Hence, the resulting predictions are difficult to explain to users. To address this problem, the contributions of this paper are threefold. First, we propose a novel stacked *two-level ensemble model* combining multiple ETA models; we show that the stacked model outperforms state-of-the-art ML models. However, the complex ensemble architecture makes the resulting predictions less transparent. To alleviate this, we investigate explainable artificial intelligence (XAI) methods for explaining the first- and second-level models of the ensemble. Third, we consider and compare different ways of combining first-level and second-level explanations. This novel concept enables us to explain stacked ensembles for regression tasks. The experimental evaluation indicates that the considered ETA models correctly learn the importance of those input features driving the prediction.

1. Introduction

In intelligent transportation systems for fleet coordination and optimization (e.g., a ridesharing service), the computation and optimization of taxi schedules are often supported by a component that estimates the duration or time of arrival for a given trip. To illustrate the problem of ETA, in Figure 1, we show two taxis Y and Z that aim to serve three passengers A, B, and C. Even in this small example, different alternative schedules are to be considered in order to find a close-to-optimal one. Using an algorithm for ETA that is independent of a route, it is possible to avoid having to compute all routes in advance, which leads to considerable speedup for larger and dynamic problem instances.

ETA also helps provide models for predicting upcoming taxi trips, e.g., *when* a taxi will pick up a passenger or *how long* a trip will take for a driver/passenger. State-of-the-art approaches have shown that high prediction precision can be achieved using ML [1–4]. A promising option to further increase prediction precision is ensemble models [5]; a special type of ensemble models is a stacked ensemble: here, the output of multiple first-level models is combined via another (second-level) model [5], for example, to the final estimation of the ETA for a taxi trip. The higher variety achieved via multiple models, potentially of different types, can better represent and interpret the diversity of the data and potentially increase prediction precision. One drawback is that by combining several black-box models with a second-level black box, the resulting model becomes even less transparent. This means that it is very difficult to understand why the model proposes a certain solution. One option to remedy this drawback is to apply XAI methods [6, 7] like Shapley additive explanations (SHAPs), which aim to explain the output of complex nonlinear ML models like

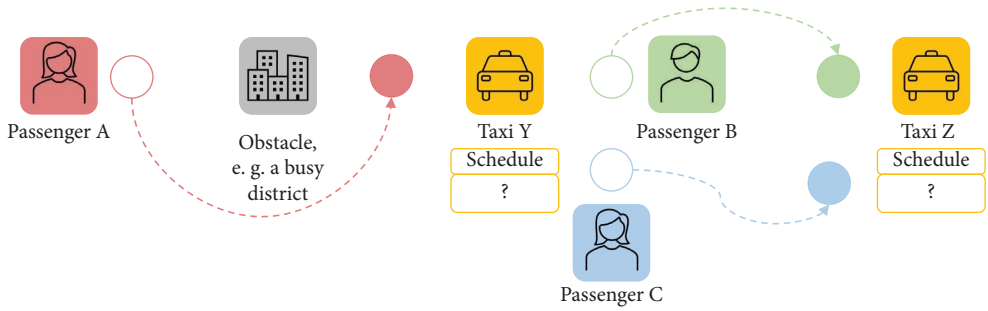

Figure 1: Motivating scenario about ETA for the planning of taxi schedules.

neural networks. With such XAI methods, we are able to learn the influence of input features on an estimated trip duration; e.g., "the hour (8:00 am) increases the estimated trip duration by 25 seconds."

The main contributions of this paper are as follows:

(1) Inspired by the good result for stacked ensembles in other problems [8–10], we propose and evaluate a stacked ensemble model for route-free estimation of trip durations.

(2) We enable explainability of stacked ensemble structures by extending existing XAI methods based on feature importance; in particular, we propose and compare three novel joining methods.

The paper is organized as follows: In Section 2, we derive our research gap and aim based on a related work review on route-free ETA and the explanation of stacked ensembles. In Section 3, we describe preliminaries, the datasets used to evaluate the prediction precision of the stacked ensemble, the first-level ETA models, the selection of the XAI methods, and the evaluation procedure. Subsequently, the construction of the stacked ensemble model is described in Section 4.1. In Section 4.2, we select and apply state-of-the-art XAI methods to explain the first-level models. Then, we propose the joining methods to explain the ensemble and conduct simulation experiments to evaluate our approach in Section 4.3. Section 4.4 discusses our findings with respect to the research aim and points out limitations as well as venues for future work; we conclude the paper in Section 5. The source code used in this paper can be accessed online via [11].

2. Related Work

2.1. Route-Free Estimated Time of Arrival. We list all the considered works about route-free ETA in Table 1. The authors of [12, 13] develop a route-free ETA approach as part of a larger ridesharing service. First, Jindal et al. [12] estimate a trip's distance and then its ETA—both via fully connected feedforward neural networks (FCNNs). Haliem et al. [13] tackle ETA based on a single FCNN with two hidden layers. While both use a relatively simple network architecture, they achieve remarkable prediction precision.

Among the approaches that focus on ETA, Wang et al. [14] propose a relatively simple neighbor-based method. Similar to [1, 12, 13], we use a FCNN, but with a different architecture. After searching for the best representation of the pickup and dropoff location that is passed into an ETA model, Schleibaum, Müller, and Sester [4] propose another FCNN architecture. Tag Elsir et al. [15] develop an advanced deep learning-based system; they incorporate both spatial-temporal and external features via convolutional, fully connected, and attention layers.

Similar to [12], which use an ensemble of two networks for two different tasks, Zou, Yang, and Zhu [16] propose a stacked ensemble of a gradient boosting decision tree and a fully connected feedforward neural network to estimate the time of arrival; both first-level models consume the same feature set.

2.2. Explaining Ensembles. As shown in Table 2, except for [17, 20, 21], all works focusing on explaining ensembles only tackle a classification problem. The majority of these works [18, 19, 24, 26] explain the ensembles post hoc by extracting rules. Given an ensemble of homogeneous models and homogeneous feature sets, Bologna and Hayashi [18] propose to transform the models of the ensemble into discretized interpretable multilayer perceptrons—a neural network derivative. From this new ensemble, rules are extracted as an explanation. In a more recent work, Bologna [19] proposes another method to extract rules from same-level ensembles. Sendi, Abchiche-Mimouni, and Zehraoui [26] learn a same-level ensemble of neural networks, transform it into one of the decision trees, and use a multiagent dialog approach to extract relatively simple rules to explain the learned classification pattern. Recently, Obregon and Jung [24] also proposed to extract simple rules from an ensemble by combining and simplifying their base trees.

Those works that do not use rule extraction to explain ensembles for classification are [22, 27]. Khalifa, Ali, and Abdel-Kader [23] propose a method to transform a learned ensemble of decision trees into a single decision tree; although they limit the prediction precision of their ensemble by ceiling the depth of their decision trees, their simplified tree remains the same prediction precision as the ensemble is remarkable. To explain a stacked ensemble for classification, Silva, Fernandes, and Cardoso [27] present the results of several XAI methods—text-based rules extracted from a decision tree, feature importance from scorecards, and an example-based method—beside each other; the authors apply their explanation approach to several ensembles used in medicine and finance.

Table 1: Related work on route-free ETA.

Reference	Usage of ML model (s)	Usage of ensemble	Ensemble type	Explanation
[12]	√	(√)	Stacked	×
[13]	√	×	—	×
[14]	×	×	—	×
[1]	√	×	—	×
[4]	√	×	—	×
[15]	√	×	—	×
[16]	√	√	Same-level	×

Table 2: Related work that explains ensembles.

Reference	Tackled problem	Ensemble type	Model types	Feature sets	Explanation type
[17]	Regression	Stacked	Heterogeneous	Homogeneous	Global, post hoc
[18]	Classification	Same-level	Homogeneous	Homogeneous	Global, post hoc
[19]	Classification	Same-level	Homogeneous	Homogeneous	Global, post hoc
[20]	Classification and regression	Same-level	Homogeneous	Homogeneous	Global, post hoc
[21]	Classification and regression	Stacked	Heterogeneous	Heterogeneous	Local/global, post hoc
[22]	Classification	Same-level	Homogeneous	Homogeneous	Local, post hoc
[23]	Classification	Same-level	Homogeneous	Homogeneous	Global, post hoc
[24]	Classification	Same-level	Homogeneous	Homogeneous	Local, post hoc
[25]	Classification	Stacked	Heterogeneous	Homogeneous	Global, post hoc
[26]	Classification	Same-level	Homogeneous	Homogeneous	Local, post hoc
[27]	Classification	Stacked	Heterogeneous	Homogeneous	Local, post hoc

Similar [27], Ren, Zhao, and Zhang [25] predict the survival rate of patients and apply the XAI method SHAP to determine the contributions of the input features for a stacked ensemble. Also in the field of medicine, Ahmed et al. [17] apply several XAI methods to an ensemble that predicts the mortality rate of patients.

Both [20, 21] explain an ensemble independent from the tackled problem. While While Deng [20] also extracts rules from the same-level ensemble of decision trees, Juraev et al. [21] apply SHAP to a stacked ensemble that does both classification and regression.

2.3. Research Gap and Aim. In general, we observe that the comparability of the aforementioned route-free ETA approaches is complicated due to the varying evaluation metrics applied, the different datasets used, and the diverse feature sets selected. Even though the feature sets selected seem to depend on the architecture applied and the promising results of [16], no previous approach tried a stacked ensemble with heterogeneous feature sets at the first level.

Except for [14], all of the aforementioned approaches proposed ML-based models for ETA. Even though such complex models are known to learn intricate patterns in the input data, none made the learned patterns transparent through explanations. Furthermore, explaining a stacked ensemble for regression is not straightforward. While most related work focuses on same-level ensembles, only four works explain a stacked ensemble [17, 21, 25, 27] and all through existing XAI methods. While the authors of [17, 25] approach the ensemble as one model, the authors of [21, 27] explain each first-level model of the ensemble separately. Both approaches hide the contribution of single models to the final decision.

Consequently, we conclude that the combination of multiple models to a stacked ensemble to perform ETA and *explain both the classical single-level models and the stacked ensemble* is an open research gap. Our research aim is twofold: First, we form a stacked ensemble model to tackle ETA. Second, we aim for an explanation method to explain such a stacked ensemble. As the explanation of ensembles through the extraction of rules is common, we will focus on feature importance methods. To limit the scope of this paper, we focus on local post hoc explanations.

3. Materials and Methods

3.1. Preliminaries

3.1.1. Estimating the Time of Arrival. Given a potential trip represented by a set of features $X = x_1, x_2, \ldots, x_n$ such as the latitude and longitude of its starting location, ETA aims to predict its duration $\hat{y} \in \mathbb{R}$ by a function f so that $f(X) = \hat{y}$. The goal is to find an f that minimizes the difference between $\hat{y}$ and the real duration y. As y is continuous, the problem described is a classical regression problem. As shown above, most of the related work uses deep learning to learn f based on a set of historical trips and their durations. Because we consider route-free ETA, information about the route, such as the number of turns on the route, and information not known before a trip starts, such as a traffic accident on the route happening after the start of a trip, are excluded from X.

3.1.2. Ensemble Learning. The function f that tackles the aforementioned ETA problem can be realized through a stacked ensemble with two levels. On the first level, such an ensemble is a composition of multiple functions $\psi_i \in \Psi$. Each ψ_i estimates f based on its feature subset $X_i \subseteq X$ so that $\psi_i(X_i) = \hat{y}_i, \forall \psi_i \in \Psi$. On the second level of the ensemble, another function ζ estimates y based on the outputs of the first level so that $\zeta(\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{|\Psi|}) = \hat{y}$. As we focus on a heterogeneous ensemble, we require the models that realize the functions $\psi_i \in \Psi$ to be of different types—like a tree—and a neural network-based model. An illustration of the ensemble architecture is shown in Figure 2.

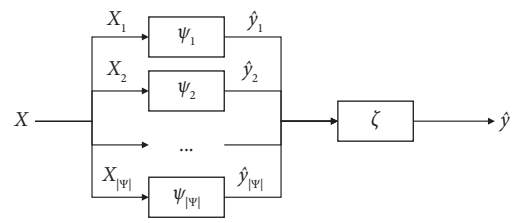

Figure 2: Architecture of a stacked ensemble with two levels—the models $\psi_1, \psi_2, \ldots \psi_{|\Psi|}$ build the first level and the model ζ the second level.

3.1.3. Explanations. Throughout this paper, we consider an explanation as a vector e of length $|X|$ assigning a value to each input feature $x \in X$ for a given prediction $\hat{y}$: $e = e_1, e_2, \ldots, e_x$; so, $e \in \mathbb{R}^{|X|}$. As we consider the prediction as given, we explain the post hoc model. To differentiate the explanations in an ensemble, we add a model superscript to an explanation: e^M.

3.2. Datasets

3.2.1. Selection and Features. We select two datasets: the New York City Yellow taxi trip data from 2015 to 2016, see [28], and one recorded in Washington DC in 2017, see [29]. We select the former because it was used several times to demonstrate an ETA approach, and it is the dataset used mainly in this paper. We additionally include the Washington DC dataset to increase the generalizability of our experiment regarding the usage of ensembles to increase the prediction precision. For both datasets, we rely on the feature engineering described by Schleibaum, Müller, and Sester [4], which makes use of or enhances the dataset by the following features: the location-based ones with (1) the pickup/dropoff as degree-based coordinates and (2) the indices of a 50-meter square grid as an alternative representation. To represent the start time of a trip, (3) the *month*, (4) *week*, (5) *weekday*, and (6) the indices of a *5-minute time bin*, which represents the hour and minute, are used. Moreover, we use (7) the *temperature* at the hour a trip starts and calculate (8) the *haversine distance* between pickup and dropoff location.

3.2.2. Outlier Removal. For removing outliers, we also use the criteria from Schleibaum, Müller, and Sester [4], and the description of the following method partly reproduces their wording. Overall, around 3% of the trips from the New York City dataset and around 19% percent from the Washington DC dataset are filtered out. A trip can be an outlier because one of its locations is not in the area studied, which is shown in Figure 3, or not in a district like erroneously being recorded in the Hudson River. Moreover, a trip's reported duration could be unreasonably low or high or could not be logically correlated with the distance between pickup and dropoff locations; we also remove trips with a distance of zero. Compared to other papers, the criteria are relatively moderate, and therefore, the comparison to approaches not reproduced is fairer.

3.2.3. Characteristics. To better understand the data, in Figure 4, we visualize the average duration of the trip per weekday for both datasets. In both, the number of trips is relatively low during the early morning; during the week, it has one peak at around 8 am and another one at around 5 pm. As expected, during the weekend, the morning peak does not exist; on average, the average duration of trips is lower than during the week.

To show the area considered and to better understand the distributions of pickup and dropoff locations, we visualize both in Figure 3. As expected for the Yellow taxis in New York City, the vast majority of trips start in Manhattan; most of the trips not starting in Manhattan begin at John F. Kennedy Airport. As for the dropoff locations, the general behavior is similar, but more trips end outside of Manhattan.

3.3. Estimated Time of Arrival Models. We take the three ETA models proposed by Schleibaum, Müller, and Sester [4] and their hyperparameters as our first-level models. We chose these because they are sophisticated ML methods previously used for tackling the problem of static route-free ETA. Furthermore, these models are based on bagging (learning multiple models from different subsets of a dataset), boosting (learning multiple models sequentially based on the previous models), and neural networks (nodes stacked in several layers enabling the capturing of complex patterns especially when trained on large datasets). Thereby, the three ML methods random forest (RF), XGBoost, and a neural network based on three diverse concepts provide a good basis for a heterogeneous ensemble.

As alternatives for the second-level model, we consider the same ML methods and add a relatively simple multiple linear regression (MLR). Regarding the main dataset or the one from New York City, we use 1M trips for training and validation from 2015 and another 250K from 2016 for testing. For the dataset from Washington DC, we use less or 600K trips for training and validation and another 50K for testing. As training data for the second-level models, we use the predictions of the first-level models on the validation data and use the same test data as before. We do not use the same training data twice or for the first- and second-level

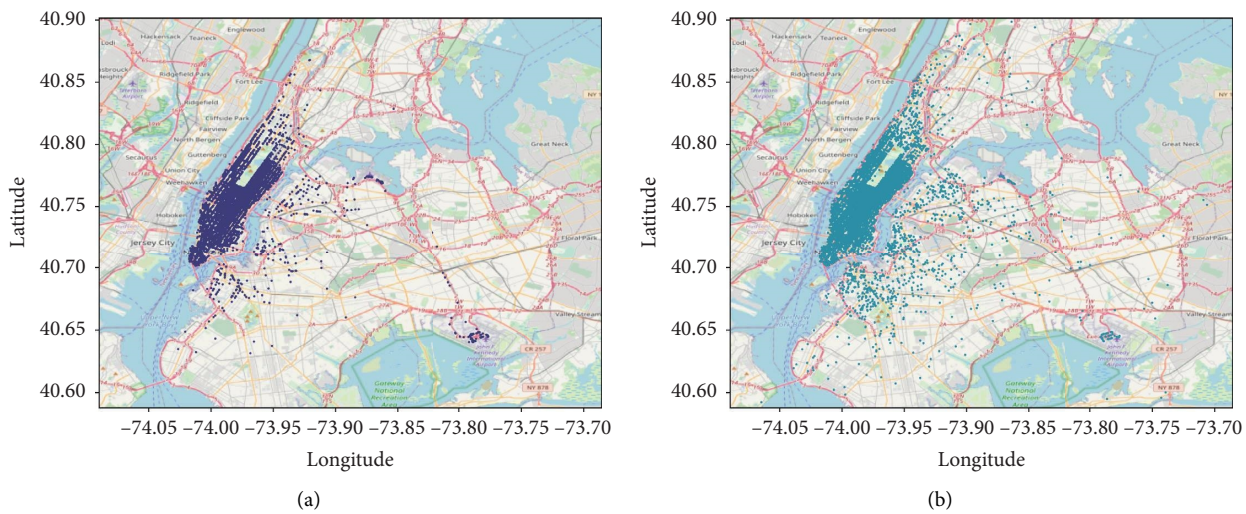

Figure 3: Distribution of the pickup (a) and dropoff (b) locations for randomly selected trips from the training data of the New York City dataset.

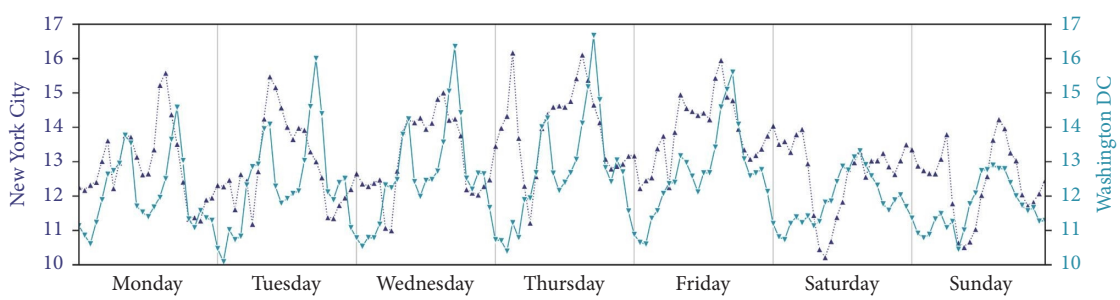

Figure 4: Distribution of the average duration in minutes over the week in the New York City dataset (darker blue) and in the Washington DC dataset (lighter blue).

models to reduce overfitting. We do not tune the hyperparameters of the second-level models and, therefore, consider not using a validation dataset as fine. Except for the MLR, which does not have any hyperparameters, for the second-level models, we use the same hyperparameters as for the first-level models. The only difference is that we decrease the number of trees for RF and XGBoost from 300 to 100 and the number of hidden layers for the neural network-based model from four to two; we choose smaller models compared to the first-level models because the number of input features or the variety of the model input is reduced substantially.

3.3.1. Baselines. We select three of the approaches presented in Table 1 reproducible from the corresponding paper as baselines [1, 12, 13]. For all three, we perform hyperparameter tuning via a random grid search. In particular, we tune the learning rate and batch size. We perform early stopping with a patience of 30 epochs. While we use the mean absolute error (MAE) for optimizing [12, 13], the mean squared error is used for reproducing [1] as described in their paper.

3.4. Selection of XAI Methods. To demonstrate our approach, we select two commonly used XAI methods—local interpretable model-agnostic explanations (LIME) and SHAP—which are described below. We chose these XAI methods because both are model-agnostic and can, therefore, be applied to all models of the heterogeneous ensemble. Moreover, both create local post hoc explanations that can be used to explain to ETA users such as taxi drivers and passengers. Although all first-level models are explained via the XAI methods, only the second-level model that performs the best will be explained.

3.4.1. Local Interpretable Model-Agnostic Explanations. Ribeiro, Singh, and Guestrin [30] present LIME, which explains predictions based on a linear surrogate model by minimizing two aspects: the goodness of the local approximation of the interpreted model in the observations neighborhood and the complexity of the surrogate model. This post hoc XAI method outputs a vector- or graphics-based explanation that is visualized differently by software libraries. The main formula presented by Ribeiro, Singh, and Guestrin [30] is

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} \mathscr{L}(f, g, \pi_x) + \Omega(g). \quad (1)$$

The importance of the feature x from a sample is extracted from the surrogate model g from all possible surrogate models G that describe the black-box model f and the neighborhood of x—denoted as π_x—best, while also minimizing the complexity of the surrogate model $\Omega(g)$.

3.4.2. Shapley Additive Explanations. Another model-agnostic XAI-method—SHAP—was proposed by Lundberg and Lee [7]. It is able to generate local explanations for a given sample by making the features' importance transparent. Therefore, SHAP utilizes the famous Shapley values from cooperative game theory. More concretely, Lundberg and Lee [7] presented the following formula:

$$\phi_i = \sum_{S \subseteq F \setminus \{i\}} \frac{|S|!\,(|F| - |S| - 1)!}{|F|!} \left[f_{S \cup \{i\}}\left(x_{S \cup \{i\}}\right) - f_S\left(x_S\right) \right]. \quad (2)$$

Here, the contribution of a feature ϕ_i is estimated by iterating over all subsets of features S of the feature set F without the feature i. The fraction in the sum weights the difference between the output of the model to be explained—represented by the function f—with and without i or the contribution of i.

3.5. Evaluation

3.5.1. Prediction Precision of ETA Models. Similar to Schleibaum, Müller, and Sester [4], we apply three evaluation metrics common for regression tasks: (1) the mean absolute error ($\text{MAE} = 1/N \sum_i |y_i - \hat{y}_i|$), which in our case returns the average error per trip in seconds, (2) the mean relative error ($\text{MRE} = \sum_i |y_i - \hat{y}_i| / \sum_i y_i$), and (3) the mean absolute percentage error ($\text{MAPE} = 1/N \sum_i |(y_i - \hat{y}_i)/y_i|$), which is robust to outliers. Because the latter two produce percentage values, they are also relatively easy to understand and put the error in perspective to a trip's duration.

3.5.2. Scenarios for Explanations. To demonstrate and evaluate our explanation approach, we randomly select ten trips from the New York City test data for four scenarios. Each scenario has two opposing characteristics that are described in the following together with the scenarios:

(1) SC1: *off city center vs. city center*: we compare trips that start outside of the city-center—a rectangle with the bottom left at coordinate (40.7975, −73.9619) and top right at (40.8186, −73.9356)—with those that do start in the city center—a rectangle with the bottom left at (40.7361, −73.9980) and top right at (40.7644, −73.9770).

(2) SC2: *night time vs. rush hour*: here, we choose some trips that start early in the morning—3 am to 5 am—and some that start during the NYC rush hour—4 pm to 6 pm.

(3) SC3: *low vs. high temperature*: in this scenario, we compare trips with a relatively low temperature—trips that are in the 0.25 quartile and not in the 0.1 decile—with trips that took place at a high temperature—trips that are in the 0.75 quartile and not in the 0.9 decile.

(4) SC4: *low vs. high distance*: we select trips with a relatively high/low distance between pickup and dropoff locations—we use the same boundaries as for SC3 for the feature *haversine distance* to select the trips.

4. Results and Discussion

4.1. Models for Estimating the Time of Arrival. We take the three ETA models proposed by Schleibaum, Müller, and Sester [4], as our first-level models as well as their hyperparameters, which have been chosen via Bayesian optimization. The first model is based on RF (*L1-RF*) with 300 trees and a maximum tree depth of 89; the number of maximal features per node is chosen automatically, and the minimum number of samples per leaf and split are set to four. The second model is based on XGBoost (*L1-XGBoost*) and also consists of 300 trees but has a maximum tree depth of eleven; the minimum number of instances required in a child is set 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. The third model is based on a FCNN (*L1-FCNN*) with four hidden layers and 300, 150, 50, and 25 corresponding neurons. Similarly to Schleibaum, Müller, and Sester [4], we set the batch size to 128, the learning rate to 0.001, train the network for 25 epochs, and select the best model along the training to minimize overfitting.

Besides the first-level models, we propose four second-level models or ensembles. Because we use all three first-level models for each ensemble, all four ensembles are heterogeneous. The first second-level model is a relatively simple one based on an MLR referred to as *L2-MLR*. The second one is based on RF (*L2-RF*) with 100 trees in the forest; for the third, XGBoost-based model or *L2-XGBoost*, we chose the same number of trees. For both *L2-RF* and *L2-XGBoost*, we do not train the hyperparameters as these methods usually achieve a high prediction precision without any hyperparameter tuning. The fourth ensemble (*L2-FCNN*) combines the output of the first-level models via a FCNN with two fully connected hidden layers—50 and 25 corresponding neurons—and otherwise similar hyperparameters to the *L1-FCNN*.

Table 3 shows that for the New York City dataset, the MAE or average prediction error in seconds per trip is around 178 seconds for the L1-FCNN and a couple of seconds higher for L1-RF and L1-XGBoost. The results for the other evaluation metrics—MRE and MAPE—are similar and put the prediction error in perspective to the trip duration. Regarding the New York City dataset and the second-level models, all models are able to outperform the first-level models with regard to MAE and MRE. Only the L2-FCNN is

Table 3: Comparison of our ETA models of the first and second level based on different evaluation metrics.

	Data set	New York city			Washington DC		
	Evaluation metric	MAE (seconds)	MRE	MAPE	MAE (seconds)	MRE	MAPE
Level 1	L1-RF	180.694	0.2158	27.8689	179.5912	0.2373	30.1512
	L1-XGBoost	183.4192	0.219	27.1137	190.2613	0.2514	30.4033
	L1-FCNN	178.2321	0.2129	23.7561	169.8152	0.2244	**24.372**
Level 2	L2-MLR	172.2439	0.2057	25.2758	171.178	0.2262	27.1985
	L2-RF	183.2319	0.2188	26.9828	183.7377	0.2428	29.5762
	L2-XGBoost	173.6526	0.2074	25.3077	172.7287	0.2283	27.5419
	L2-FCNN*	**169.4285**	**0.2023**	**22.9121**	**167.9959**	**0.222**	24.6133
Baselines	[12]	185.9265	0.2256	23.8429	181.1275	0.2374	27.4261
	[1]	201.5998	0.2455	28.1508	203.8581	0.2673	35.4898
	[13]	185.3999	0.2257	28.3598	174.3907	0.2286	25.7570

*This prediction precision is better than the one presented by Schleibaum et al. [4].

able to outperform all first-level models in all evaluation metrics with an MAE of 169 seconds or an MRE of around 20 percent. Interestingly, the L2-MLR achieves a remarkable prediction precision that is better than that of L2-RF and similar to that of L2-XGBoost. For the models trained and tested on the Washington DC dataset, we observe that on the first level, the L1-FCNN with an MAE of around 169 seconds is able to outperform L1-RF by 10 seconds and L1-XGBoost by 20 seconds. Regarding second-level models, we observe that except for L2-RF, all models achieve a remarkable prediction precision. In contrast to the models trained and tested on the New York City dataset, none of the second-level models is able to outperform the mean absolute percentage error (MAPE) achieved by the L1-FCNN.

As shown in the last three lines of Table 3, we outperform the baseline approaches by at least 15 seconds on the New York City dataset and by at least 6 seconds on the Washington DC dataset when considering the MAE. The results are similar for the other two evaluation metrics.

4.2. Explaining the First-Level Estimated Time of Arrival Models. In Figure 5, we visualize the explanations generated by LIME per scenario, its characteristics, feature, and ETA model. Each triangle represents the importance of a feature for one trip from a scenario—the lighter triangles are from the first or "lower" characteristic of the scenario. The lines connect the importance of the features for one sample or trip. Concrete values can be interpreted as follows: for instance, the left-most triangle in SC1—(a) of Figure 5—has a value of around −575. This refers to a relatively strong negative influence of the concrete distance value on the corresponding estimated duration of the trip. This most likely refers to a trip with a small distance between the pickup and dropoff locations.

In SC2—top right—and SC4—bottom right—the two characteristics of each scenario are visually separated for the features that constitute the scenario—the 5-minute time bin for SC2 and the distance for SC4. As expected, the other features are not separable because they are more or less randomly distributed over the space of each feature—not completely random, as some features might slightly correlate with the feature of interest. Moreover, the separation is in the correct order, which means that the "lower" characteristic also has a lower importance of features than the "higher" characteristic of the scenario. Therefore, the ETA models appear to have properly learned the expected behavior in these scenarios. Even though for the majority of trips in SC1, the difference between the pickup off city center and in city center is learned, and some trips of the "lower" and "higher" characteristics interfere. For instance, for *pickup X*, the *L1-FCNN* feature importance of both scenario characteristics overlaps. As regards SC3, we observe that the reported feature importance for the temperature is low or close to zero. While this could indicate that the ETA models have not learned the underlying pattern, similar to Schleibaum et al. [4], we argue that the overall feature contribution of the weather or temperature is low.

While the concrete values or feature contributions generated via SHAP differ from the feature importance of LIME, we observe similar results in Figure 6. For SC2 and SC4, the two characteristics of the scenarios are visually separated only by the feature of interest; the separation for SC1 and SC3 is not clear.

4.3. Explaining the Ensemble Model for Estimated Time of Arrival. In the following, we describe three relatively simple but novel methods to join the explanations from the first and second levels. To be able to compare the outputs for all three joining methods and samples in a better way, we first normalize the second-level explanation per sample so that they sum up to one. The joining methods (JMs) are as follows:

(1) (JM1: *adding a dimension*) Here, we simply output both the first and second-level explanations—which is meant by "additional dimension"—simultaneously. Therefore, we weigh the feature contribution or importance of a feature at the first level with the contribution or importance of the prediction at the second level. Consequently, we join the first and second-level explanations without losing any information.

(2) (JM2: *basic join of the contributions*) To determine the contribution or importance of a feature, we

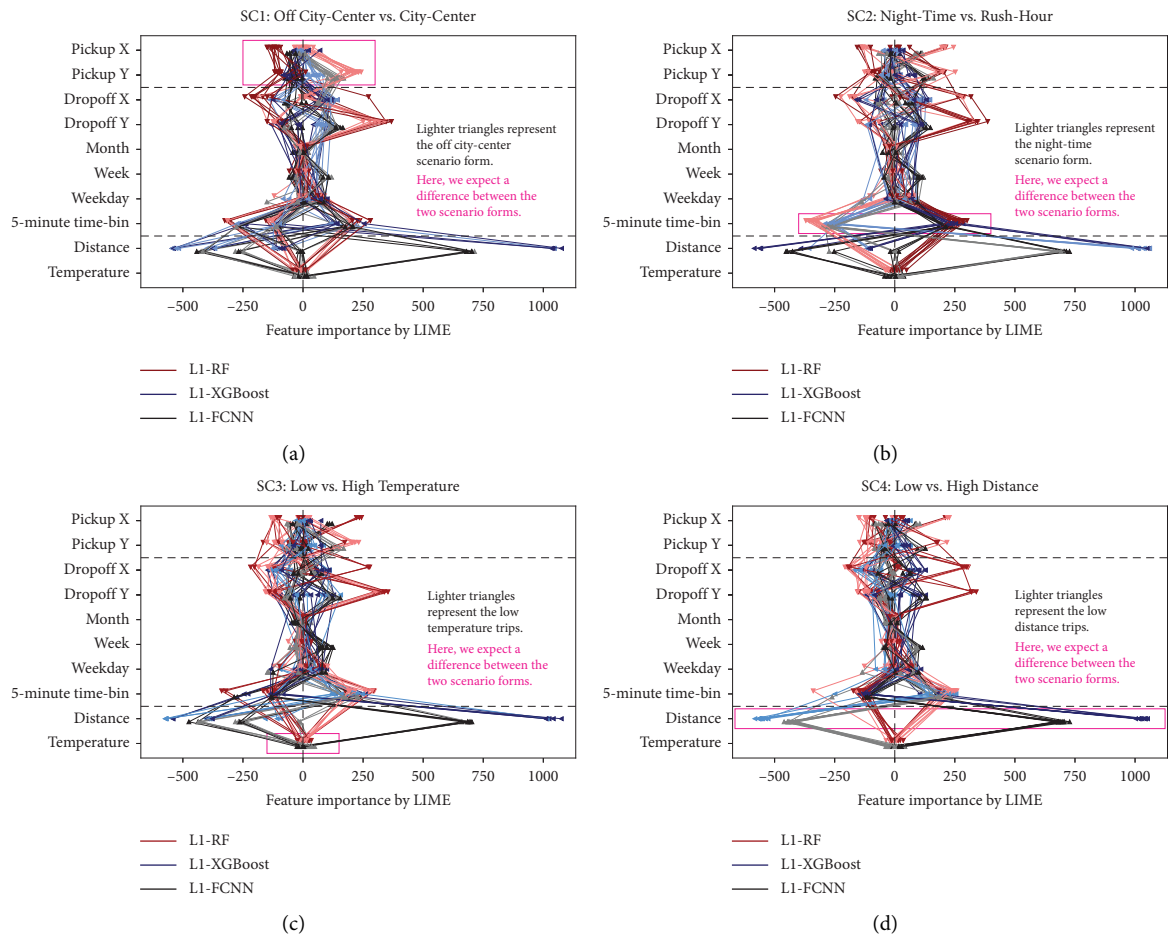

Figure 5: Local feature importance via LIME per feature of the samples in the scenarios for the first-level models; each plot refers to one scenario: (a) for SC1, (b) for SC2, (c) for SC3, and (d) for SC4; the ten trips with an expected lower influence are marked with lighter triangles—the ones with an expected higher influence with less light triangles; each line connects the feature importance for one trip along the various features used by the corresponding model.

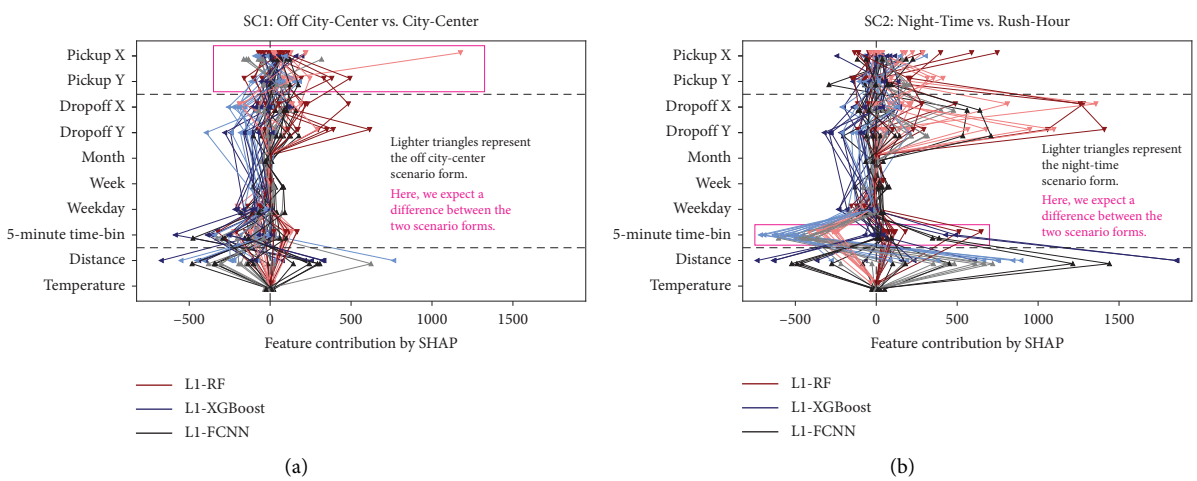

Figure 6: Continued.

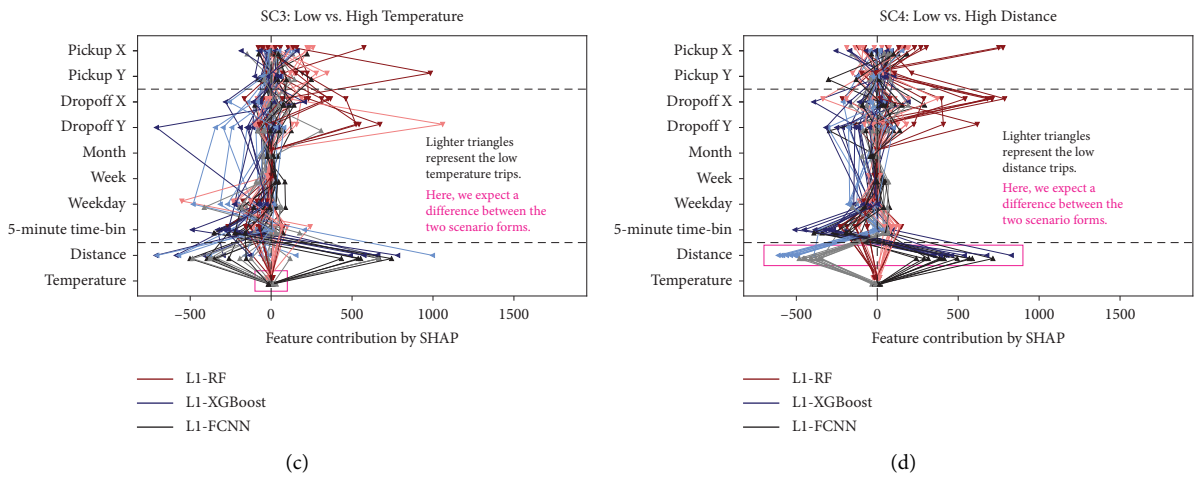

Figure 6: Explanations via SHAP per feature, sample, and scenario for the first-level models; each subfigure refers to one scenario: (a) for SC1, (b) for SC2, (c) for SC3, and (d) for SC4; the ten trips with an expected lower influence are marked with lighter triangles—the ones with an expected higher influence with fewer light triangles.

compute the dot product between the vector that contains the contribution or importance of that feature for each first-level model and the vector that contains the contribution or importance of each first-level model on the second level; the product is then the joint contribution or importance of that feature for a given sample.

(3) (JM3: *diversifying the contributions*) Here, we use the result from JM2 as a basis and define a threshold α, which is the mean value of the distribution or influence of each first-level model on the second level or, e.g., 1/3 with our three first-level models. Next, every value below that threshold is reduced by a value β to be increased by the collected value in the next step. If the values cannot be reduced by β, because they would become negative, only the difference to zero is used and redistributed to the values above α. Thus, the second-level influence is diversified. In the following, we set β to 0.5 or relatively high, as the number of first-level models is only three.

All three joining methods are compared to a *baseline (BL)* method. This method generates explanations by explaining a function that wraps the whole ensemble. Within this function, features that are an alternative representation of other features, like the X-index of a 50-meter square grid, are also generated within that function from the corresponding base feature, i.e., the latitude of the pickup location. In Figure 7, we show an overview of our three joining methods compared to the baseline.

In Figure 8, we show the feature importance for LIME joined via all three proposed joining methods as well as for the BL. Similar to previous findings, in each graphic of Figure 8, for each trip and method, a line—or three for JM1—is shown, this time without triangles. As we only want to demonstrate the joining methods, we omit all scenarios except for SC2 here, but we show the corresponding graphics for SC1, SC3, and SC4 in Figure S1 in Supplementary Material.

In Figure 8, the difference between JM1 and BL/JM2/JM3 is obvious: JM1 shows much more information, including the proof that all three first-level models are used by the *L2-FCNN*. Even though the two scenario characteristics have the expected difference at the 5-minute time bin, verifying the difference among the first-level models is hard for JM1. Regarding JM2, we observe a relatively high difference in the BL joining method as is for instance visible in the feature importance of the 5-minute time bin or the distance of the night-time characteristic of the scenario. As expected, the JM3 makes the smaller feature importance values smaller and the larger ones larger, thereby diversifying the feature importance along all features slightly.

When applying the joining methods to the SHAP values for the same scenario, as shown in Figure 9, we observe similar results. While the difference between the night-time and rush-hour characteristic of SC2 is visible for all joining methods, this time JM2 and JM3 in general reduce the feature importance. This is in contrast to the explanations generated by LIME.

In Figure 10, we visualize the Shapley values for the features used to build the scenarios via the joining methods JM2 and JM3 per scenario and their two opposing characteristics—"lower" and "higher"—to further investigate the differences to the BL; JM1 is omitted in the figure as it is hard to compare in the visualized regard. As expected, the Shapley values generated via JM2 and JM3 do not vary much compared to the BL; like for the 5-minute time bin and SC2H, JM2 and JM3 slightly change the Shapley values in the positive direction. For the distance and SC4L, the Shapley values are moved in the opposite direction. In general, the difference expected in the scenarios gets slightly smaller, but it is still clearly shown. A similar figure for LIME can be found in Figure S3 Supplementary Material.

4.4. Discussion

4.4.1. Ensemble for ETA. We developed multiple alternatives to combine the outputs of the RF, XGBoost, and FCNN models via another model. Even though several second-level

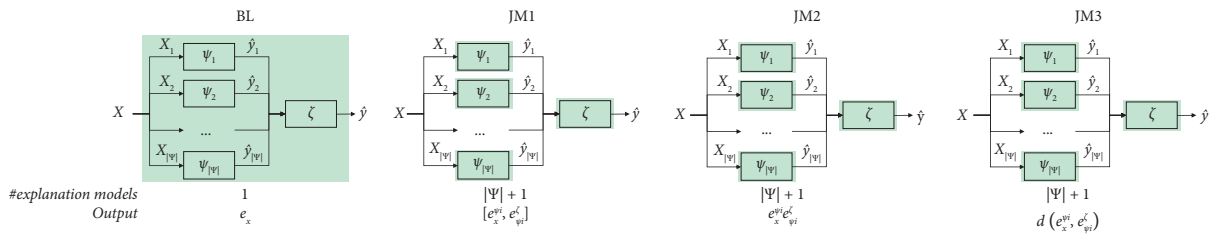

Figure 7: Overview of our three joining methods (JM1, JM2, and JM3) compared to the baseline (BL). Those models to which an explanation method is applied are highlighted in green, showing that the number of explanation models is higher (providing more insights) in our proposed explanation methods. Furthermore, we show the output—e refers to an explanation and $d()$ to the diversifying function described previously.

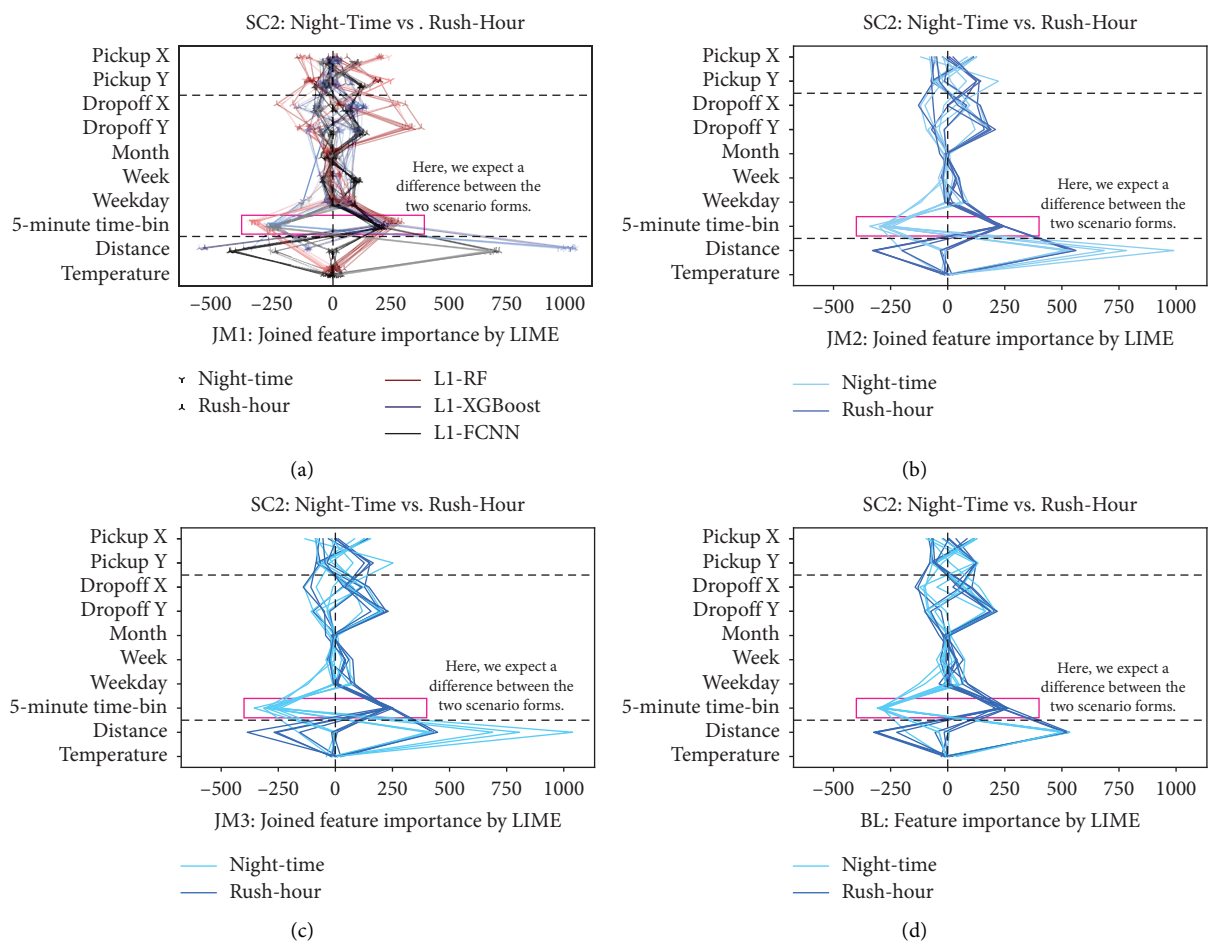

Figure 8: Joint local feature importance via LIME for each feature of the samples in the second scenario (SC2: night time vs. rush hour) for the joining methods JM1 (a), JM2 (b), and JM3 (c) compared to the BL (d). Each line connects the feature importance for one trip, and (a) the line width corresponds to the influence on the second level.

models achieved a high prediction precision on the New York City dataset, only an FCNN-based one was able to outperform our previous models from Schleibaum et al. [4] in all evaluation metrics. Interestingly, for the Washington DC dataset, the results were not that clear: while the FCNN-based ensemble performed better than the first-level FCNN-based model as regards the MAE and mean relative error (MRE), for the MAPE, the observed pattern is the opposite. We believe that this is caused by three reasons. First, feature selection and hyperparameter tuning were performed for the New York City dataset and consequently not optimal for the Washington DC dataset. Second, the Washington DC dataset with around 650K trips used is much smaller than the New York City one with 1.25M trips, causing the second-level models to have much fewer data to be trained on. Third, the gap between the performance of the three first-level models is much closer for the models trained on the New York City dataset than for those first-level models trained on

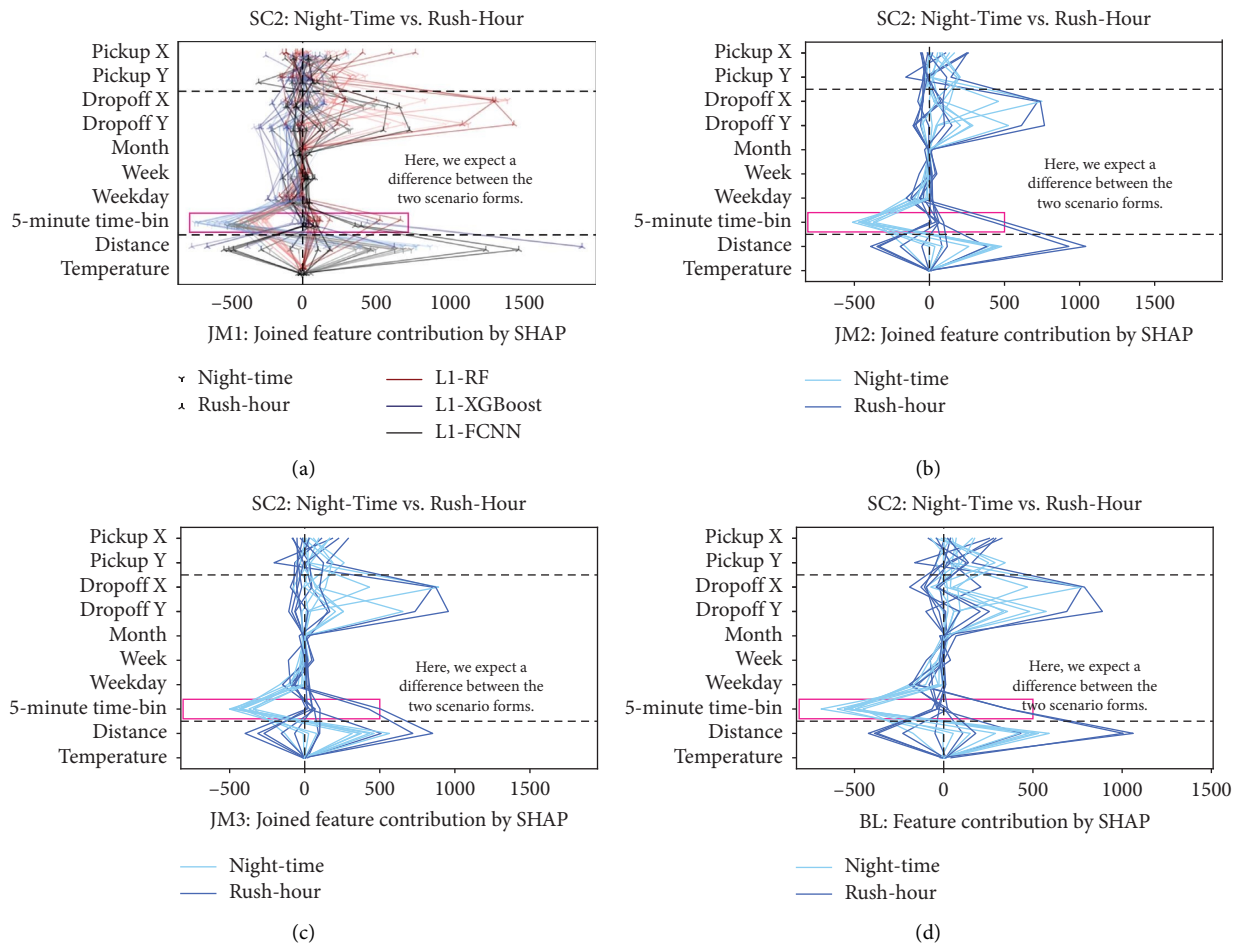

Figure 9: Joint local feature importance via SHAP for each feature of the samples in the second scenario (SC2: night time vs. rush hour) for the joining methods JM1 (a), JM2 (b), and JM3 (c) compared to the BL (d). Each line connects the feature importance for one trip, and (a) the line width corresponds to the influence on the second level.

the Washington DC dataset. Therefore, we assume that a better-performing XGBoost model or excluding it from the ensemble could further improve the prediction precision. As we already outperformed the approaches of [1, 12, 13] in our previous work [4], we consider the usage of a stacked heterogeneous ensemble as an effective method to increase the prediction precision for static route-free ETA. With the dataset considered, we reduced the MAE by nine seconds to around 169 seconds per trip on average; both MRE and MAPE were reduced by around one percentage point.

4.4.2. Explaining First-Level ETA Models. We applied the two model-agnostic XAI methods LIME and SHAP to evaluate and explain our first-level ETA models post hoc and locally. In SC2 and SC4, we could show that all three models learned the expected behavior. For SC3, all ETA models that include the temperature consistently learned a low influence of the temperature on the ETA. As described previously, this is most likely caused by the low influence of weather-related data in general rather than a pattern that is not properly learned by our ETA models. Even though Schleibaum et al. [4] showed the positive influence of including the feature in the models on the prediction precision, as their general influence is low, the explanation or feature importance value assigned is not very meaningful. In case someone focuses on explainable ETA models, removing the temperature or the month feature might be worth considering. Regarding SC1, we could show that information like the pickup location, which is encoded into multiple and, therefore, correlating features, is difficult to explain by LIME and SHAP. We observed that the explanations produced by LIME are more separated in our scenarios than those of SHAP. As the focus of our work is not to compare LIME and SHAP, we refer the interested reader to the work of Belle and Papantonis [31]; but in general, LIME has a relatively low runtime and SHAP has the advantage of producing additive explanations.

4.4.3. Joint Explanation of Ensembles for ETA. We presented three relatively simple methods for joining the first- and second-level explanations of an ensemble to generate a joint explanation. The main advantage and at the same time drawback of joining method JM1 (*adding a dimension*) is that more information or all explanations are shown. When—as we did—multiple trips are shown in one graphic,

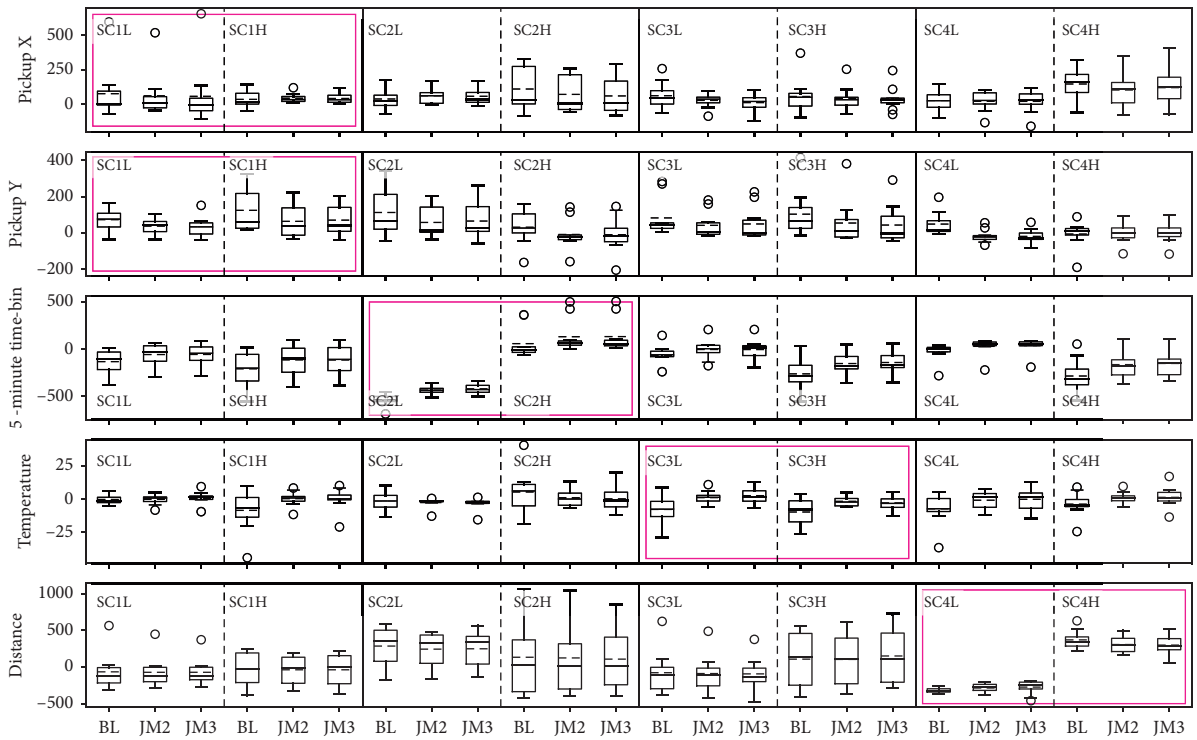

Figure 10: Box plot per feature—those affected by the scenarios like the 5-minute time bin for SC2—and joining method compared to the BL for each scenario in its lower (e.g. SC1L) and its higher (e.g. SC1H) characteristic; the dashed lines in the boxes of each box plot are the mean values, and the pink rectangles mark the features of the scenarios.

we assume that it is harder to understand, but on the other hand, especially when only one trip is visualized, this provides additional insights not provided by JM2 and JM3. For instance, it might be interesting to see if different first-level models disagree on a feature's importance for a specific sample, how strong the influence of each of the models is, and if some relation was not learned correctly. For such a case, a smarter choice of colors or an alternative to the used line plots could improve understandability. However, with respect to larger ensembles or those that have more or many first-level models, less dense explanation created by JM1 might be confusing or not understandable anymore.

As regards joining method JM2 (*basic join of the contributions*), we observed an unexpectedly high difference in the BL method. We believe that this difference is at least partly caused by correlated features such as the pickup latitude and longitude. Nevertheless, the general direction of the feature importance is similar. Even though the XAI methods might not be built for correlated features, especially in stacked ensembles and practice, correlated features exist. Interestingly, when considering LIME, the larger values are made larger; for SHAP, the effect is opposite. For JM3 (*diversifying the contributions*), we observed the same but slightly stronger effect. In contrast to JM1, JM2, and BL, JM3 has a hyperparameter that has to be chosen by the user, which makes this method more complicated to apply.

Interestingly, none of the related work [22, 27, 32] has used the BL method to generate an explanation or compare their explanation to it. As we did not find other literature regarding explaining ensembles, we assume that our work presents three novel joining methods. While the general concept that we applied to create a joint explanation of a stacked ensemble with two levels that performs a regression is relatively simple, the proposed concept is neither specific to the underlying XAI method nor to the regression models. It could even be applied to the probabilities generated by classification models. In addition, the concept does not depend on the number of first-level models and can, as we did, be applied to first-level models that only share a part of their input features. Also, the joining methods are model-agnostic, and a combination of different XAI methods is possible. When, for instance, considering one or multiple complex models on the first level, explaining them with an XAI method that has a faster inference time, and combining that on the second level with an XAI method like SHAP, is possible.

4.4.4. Limitations and Future Work. As argued before, we applied relatively moderate criteria for identifying outliers before training various ETA models. We did this to make the comparison to nonreproduced papers fairer. However, we expect that we could further increase the prediction precision of the composed ensemble model. Another option to potentially achieve a higher prediction precision is to include other ETA models into the ensemble as additional first-level models, for instance [1–3, 13].

While the evaluation of the performance of ETA models is relatively straightforward, the evaluation of explanations is not; especially, determining the influence of the slight

differences between our joining methods is affected by this problem. Moreover, we considered only four self-chosen scenarios to demonstrate and evaluate the generated explanations; many more scenarios like the ones that combine features—for instance, pickup at the city center during rush hour—might be interesting and valuable for evaluation. While we focused on generating explanations in a general way so that others can adapt and build upon our work, correlated features or information that span over multiple features like the pickup location could be explained better when the features are explained jointly or the x and y values of the pickup location and their influence on the ETA are visualized on a map.

Regarding explanations, future work will look into ways to explain information that spans over multiple features. Another option to extend this work is to use other XAI methods or use different ones for different models to generate more accurate explanations per model type. The latter could also be used to generate explanations relatively fast, for instance, by using LIME on first-level models with a greater feature space and SHAP on the second-level models with a smaller feature space. Moreover, the explanations generated here are vectors of values and, therefore, still hard to understand by affected users like taxi drivers or passengers. The explanations could be translated into more human-friendly ones, for instance, by linking an explanation of the locations influence on the ETA to points of interest like the main train station that is close to the dropoff location and thus possibly increasing the ETA for an upcoming trip. Moreover, our explanation could be enhanced from route-free to route-based ETA as such approaches are more likely to be used by taxi drivers and passengers thanks to their increased prediction precision. In addition, using the generated explanations might be beneficial not only for users of an ETA model but also for the designers of such models. Based on the explanations, some first-level models or features used in a model might be excluded—leading to a smaller and more precise ETA model.

5. Conclusions

On-demand transportation modes, such as ridesharing or ridehailing, are key building blocks of sustainable passenger transportation. Estimating the time of arrival of vehicles (taxis) in ridesharing or ridehailing is relevant for the comparison and computation of schedules and provides important information to drivers and passengers. In this paper, we investigate how the prediction precision of ETA algorithms can be improved by combining multiple ML models into a stacked heterogeneous ensemble—which, on its own, is novel and has been shown to outperform state-of-the-art static route-free ETA methods on two datasets. Furthermore, to enable the explainability and transparency of the stacked model, we proposed XAI methods for explaining the first-level models of the ensemble, as well as three novel methods for joining the first and second-level explanations of the ensemble model. To demonstrate the feasibility and benefit of our approach, we use a taxi trip dataset collected in New York City to evaluate our explanations against a baseline model that wraps the whole ensemble in one function. Based on the limitations, more tuning of the ensemble models and the inclusion of other ETA models from the related work is promising. In addition, we want to explore the explanation of correlated features and the combination of different XAI methods to explain ensembles.

Data Availability

As described in Section 3.2, we use the New York City Yellow taxi trip data from 2015 to 2016, which is publicly available, and the Washington DC taxi trip data recorded in 2017 to support the findings of this study. Links to the datasets are included in [28, 29]. Methods used to enhance the datasets by the additional features considered throughout this paper are provided in our code repository, see [11].

Disclosure

A preprint has previously been published [33].

Conflicts of Interest

The authors declare that there are no conflicts of interest with respect to the publication of this article.

Acknowledgments

We thank Steven Minich, Helene Nicolai, and Julian Teusch for providing helpful feedback, especially regarding the explanation part of the paper. 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. Open Access funding was enabled and organized by Projekt DEAL.

Supplementary Materials

In Figure S1, we visualize the LIME explanations generated via the proposed joining methods for SC1, SC3, and SC4; in Figure S2, we do the same for Figure 13: the SHAP explanations. Figure 13 shows the content of Figure 10 for LIME. Figure S1: Content of Figure 8 for for SC1, SC3, and SC4. Figure S2: content of Figure 9 for SC1, SC3, and SC4. Figure S3: content of Figure 10 for LIME. (*Supplementary Materials*)

References

[1] A. C. de Araujo and E. Ali, "Deep neural networks for predicting vehicle travel times," in *Proceedings of the 2019 IEEE SENSORS*, pp. 1–4, IEEE, Montreal, QC, Canada, October 2019.

[2] K. D. Kankanamge, Y. R. Witharanage, C. S. Withanage, M. Hansini, D. Lakmal, and U. Thayasivam, "Taxi trip travel time prediction with isolated XGBoost regression," in *Proceedings of the 2019 Moratuwa Engineering Research*

Conference (MERCon), pp. 54–59, IEEE, Moratuwa, Sri Lanka, July, 2019.

[3] Y. Li, K. Fu, ZhengWang, C. Shahabi, J. Ye, and Y. Liu, "Multi-task representation learning for travel time estimation," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1695–1704, ACM, London, UK, July, 2018.

[4] S. Schleibaum, J. P. Müller, and M. Sester, "Enhancing expressiveness of models for static route-free estimation of time of arrival in urban environments," *Transportation Research Procedia*, vol. 62, pp. 432–441, 2022.

[5] M. A. Ganaie, M. Hu, A. K. Malik, M. Tanveer, and P. N. Suganthan, "Ensemble deep learning: a review," *Engineering Applications of Artificial Intelligence*, vol. 115, Article ID 105151, 2022.

[6] P. Linardatos, V. Papastefanopoulos, and S. Kotsiantis, "Explainable ai: a review of machine learning interpretability methods," *Entropy*, vol. 23, no. 1, p. 18, 2020.

[7] M. L. Scott and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proceedings of the 31st International Conference on Neural Information Processing Systems. NIPS'17*, pp. 4768–4777, Curran Associates Inc, Red Hook, NY, USA, July, 2017.

[8] Y. Xia, K. Chen, and Y. Yang, "Multi-label classification with weighted classifier selection and stacked ensemble," *Information Sciences*, vol. 557, no. 2021, pp. 421–442, 2021.

[9] M. Gour and S. Jain, "Automated COVID-19 detection from X-ray and CT images with stacked ensemble convolutional neural network," *Biocybernetics and Biomedical Engineering*, vol. 42, no. 1, pp. 27–41, 2022.

[10] M. S. Akhtar, A. Ekbal, and E. Cambria, "How intense are you? Predicting intensities of emotions and sentiments using stacked ensemble [application notes]," *IEEE Computational Intelligence Magazine*, vol. 15, no. 1, pp. 64–75, 2020.

[11] S. Schleibaum, "Estimated time of arrival," https://gitlab.tu-clausthal.de/ss16/stacked-etaand-explanation.Mar.2022.

[12] I. Jindal, Z. T. Qin, X. Chen, M. Nokleby, and J. Ye, "Optimizing taxi carpool policies via reinforcement learning and spatio-temporal mining," in *2018 IEEE International Conference on Big Data (Big Data)*, pp. 1417–1426, IEEE, Seattle, WA, USA, December, 2018.

[13] M. Haliem, G. Mani, V. Aggarwal, and B. Bhargava, "ADistributedModel-FreeRide- sharing approach for joint matching, pricing, and dispatching using deep reinforcement learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 12, pp. 7931–7942, 2021.

[14] H. Wang, X. Tang, Y.-H. Kuo, D. Kifer, and Z. Li, "A simple baseline for travel time estimation using large-scale trip data," *ACM Transactions on Intelligent Systems and Technology*, vol. 10, no. 2, pp. 1–22, 2019.

[15] M. Alfateh, T. Elsir, A. Khaled, P. Wang, and Y. Shen, "JSTC: travel time prediction with a joint spatial-temporal correlation mechanism," in *Journal of Advanced Transportation*, L. Sun, Ed., vol. 2022, Article ID 1213221, 16 pages, 2022.

[16] Z. Zou, H. Yang, and A.-X. Zhu, "Estimation of travel time based on ensemble method with multi-modality perspective urban big data," *IEEE Access*, vol. 8, pp. 24819–24828, 2020.

[17] Z. U. Ahmed, K. Sun, M. Shelly, and L. Mu, "Explainable artificial intelligence (XAI) for exploring spatial variability of lung and bronchus cancer (LBC) mortality rates in the contiguous USA," *Scientific Reports*, vol. 11, no. 1, Article ID 24090, 2021.

[18] B. Guido and Y. Hayashi, "A comparison study on rule extraction from neural network en- sembles, boosted shallow trees, and SVMs," *Applied Computational Intelligence and Soft Computing*, vol. 2018, Article ID 4084850, 20 pages, 2018.

[19] B.. Guido, "Transparent ensembles for covid-19 prognosis," in *Machine Learning and Knowledge Extraction*, A. Holzinger, P. Kieseberg, A. M. Tjoa, and E. Weippl, Eds., vol. 12844, pp. 351–364, Springer International Publishing, Berlin, Germany, 2021.

[20] H. Deng, "Interpreting tree ensembles with inTrees," *International Journal of Data Science and Analytics*, vol. 7, no. 4, pp. 277–287, 2019.

[21] F. Juraev, S. El-Sappagh, E. Abdukhamidov, F. Ali, and T. Abuhmed, "Multilayer dynamic ensemble model for intensive care unit mortality prediction of neonate patients," *Journal of Biomedical Informatics*, vol. 135, Article ID 104216, 2022.

[22] A. Kallipolitis, K. Revelos, and I. Maglogiannis, "Ensembling EfficientNets for the classification and interpretation of histopathology images," *Algorithms*, vol. 14, no. 10, p. 278, 2021.

[23] F. Khalifa, A. Ali, and H. Abdel-Kader, "Improved version of explainable decision forest: forest-based tree," *IJCI International Journal of Computers and Information*, vol. 0, no. 0, p. 0, 2022.

[24] J. Obregon and J.-Y. Jung, "RuleCOSI+: rule extraction for interpreting classification tree ensembles," *Information Fusion*, vol. 89, pp. 355–381, 2023.

[25] N. Ren, X. Zhao, and X. Zhang, "Mortality prediction in ICU using a stacked ensemble model," in *Computational and Mathematical Methods in Medicine*, M. E. Fantacci, Ed., vol. 2022, Article ID 3938492, 12 pages, 2022.

[26] N. Sendi, N. Abchiche-Mimouni, and F. Zehraoui, "A new transparent ensemble method based on deep learning," *Procedia Computer Science*, vol. 159, pp. 271–280, 2019.

[27] S. Wilson, K. Fernandes, and J. S. Cardoso, "How to produce complementary explanations using an ensemble model," in *Proceedings of the 2019 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, IEEE, Budapest, Hungary, July, 2019.

[28] City of New York, "TLC trip record data," 2019, https://www.nyc.gov/site/tlc/about/tlc-trip-record-data.page.

[29] Kaggle, "DC taxi trips," https://www.kaggle.com/c/nyc-taxi-trip-duration.

[30] M. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?: explaining the predictions of any classifier," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pp. 97–101, Association for Computational Linguistics, San Diego, CA, USA, December, 2016.

[31] V. Belle and I. Papantonis, "Principles and practice of explainable machine learning," *Frontiers in Big Data*, vol. 4, Article ID 688969, 2021.

[32] B. Rozemberczki and R. Sarkar, "The Shapley value of classifiers in ensemble games," in *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 1558–1567, Association for Computing Machinery, New York, NY, USA, June, 2021.

[33] S. Schleibaum, J. P. Müller, and M. Sester, "An explainable stacked ensemble model for static route-free estimation of time of arrival," 2022, https://arxiv.org/abs/2203.09438.