Customized Uncertainty Quantification of Parking Duration Predictions for EV Smart Charging

Kaleb Phipps, Karl Schwenk, Benjamin Briegel, Ralf Mikut, Veit Hagenmeyer *Member, IEEE*

***Abstract*—As Electric Vehicle (EV) demand increases, so does the demand for efficient Smart Charging (SC) applications. However, SC is only acceptable if the EV user's mobility requirements and risk preferences are fulfilled, i.e. their respective EV has enough charge to make their planned journey. To fulfill these requirements and risk preferences, the SC application must consider the predicted parking duration at a given location and the uncertainty associated with this prediction. However, certain regions of uncertainty are more critical than others for user-centric SC applications, and therefore, such uncertainty must be explicitly quantified. Therefore, the present paper presents multiple approaches to customize the uncertainty quantification of parking duration predictions specifically for EV user-centric SC applications. We decompose parking duration prediction errors into a critical component which results in undercharging, and a non-critical component. Furthermore, we derive quantile-based security levels that can minimize the probability of a critical error given a user's risk preferences. We evaluate our customized uncertainty quantification with four different probabilistic prediction models on an openly available semi-synthetic mobility data set and a data set consisting of real EV trips. We show that our customized uncertainty quantification can regulate critical errors, even in challenging real-world data with high fluctuation and uncertainty.**

***Index Terms*—Smart Charging, Uncertainty, Parking Duration, Probabilistic Predictions**

ACRONYMS

BRR	Bayesian Ridge Regression
CDF	Cumulative Distribution Function
EV	Electric Vehicle
GPR	Gaussian Process Regression
IoT	Internet of Things
NGBoost	Natural Gradient Boosting
NN	Neural Network
PDF	Probability Density Function
SC	Smart Charging

NOTATION

$\mathbb{E}[\dots]$	Expected Value of Expression
$\mathcal{E}_{t,\text{PI}}$	Error Outside the Prediction Interval
$\mathcal{E}_{\text{PI}}$	Mean Error Outside the Prediction Interval
$\mathcal{E}_{\text{Tot}}$	Mean Absolute Error of the Parking Duration Prediction
$\mathcal{E}_{\text{c}}$	Critical Error of the Parking Duration Prediction
$\mathcal{E}_{\text{nc}}$	Non-Critical Error of the Parking Duration Prediction
$\mathcal{E}_{\text{INT}}$	Integral Error Between the Predicted and Empirical CDF
f_Y	Probability Density Function of Random Variable Y
F_Y	Cumulative Distribution Function of Random Variable Y
g	Arbitrary Prediction Model
$\hat{I}^{(\beta)}$	Prediction Interval With Nominal Coverage Rate $1-\beta$
$\mathcal{L}_{\text{pinball},\alpha}$	Pinball Loss for the α-Quantile
$\mathbb{P}[\dots]$	Probability of Expression
$\mathcal{T}$	Tolerance for the Non-Critical Error
$\mathcal{W}$	Average Width of the Prediction Interval
X	Exogenous Variables Used in Prediction
Y	Random Variable Modeling Parking duration
y	Observed Parking Duration
$\hat{y}$	Predicted Parking Duration
$\hat{y}^{(\alpha)}$	Predicted α-Quantile for Parking Duration
$\hat{y}^{(\underline{\alpha})}$	Predicted Lower Quantile for Prediction Interval
$\hat{y}^{(\bar{\alpha})}$	Predicted Upper Quantile for Prediction Interval
α	Used to Define the α-Quantile
η	Security Level in Percent
$\hat{\Theta}$	Estimated Model Parameters

I. INTRODUCTION

MITIGATING climate change is a major global challenge and, as observed in our previous conference paper [1], Electric Vehicles (EVs) could play a major role in reaching important climate targets, especially when charged by a highly renewable energy mix [2]. However, coupling this energy mix with an increased share of EVs causes new strains on our electrical system and can lead to grid instabilities [3]. As a result, coordinated and intelligent charging approaches, so-called Smart Charging (SC), of EVs are required [3]–[5]. These intelligent charging approaches involve integrating EVs into a smart Internet of Things (IoT) electrical grid, enabling bi-directional communication to manage power flow, and optimizing charging schedules [6]. However, these SC approaches cannot be allowed to inconvenience the user by, e.g., resulting in extra charging stops due to insufficient state of charge or forcing the user to charge at an unknown destination. Therefore, SC applications can only be successfully applied if information regarding a user's mobility behavior is available and combined into the SC application [7]. This mobility behavior includes common destinations, travel frequency, distance traveled, and how long a user stays at a specific location [7]. Additionally, for SC to be fully accepted, this mobility behavior should be integrated into the SC application without the user manually feeding parameters, such as planned parking

Manuscript received 2023-04-14. *(Corresponding Author: Kaleb Phipps)*

Kaleb Phipps, Ralf Mikut and Veit Hagenmeyer are with the Institute for Automation and Applied Informatics, Karlsruhe Institute of Technology, Hermann-von-Helmholtz-Platz 1, Karlsruhe, Germany (e-mail: kaleb.phipps@kit.edu; ralf.mikut@kit.edu; veit.hagenmeyer@kit.edu).

Karl Schwenk and Benjamin Briegel are with Mercedes-Benz AG, Sindelfingen, Germany (e-mail: karl.schwenk@mercedes-benz.com; benjamin.briegel@mercedes-benz.com).

duration, next destinations, or risk preferences, into the SC algorithm [8]. As a result, such mobility behavior must be automatically predicted.

Although mobility behavior typically follows certain patterns, there is always an aspect of this behavior that is random and unpredictable [9]. For example, a EV user may leave for work every morning at a regular time, but due to fluctuating traffic conditions or unforeseen vehicle problems, the trip duration varies [10], [11]. Similarly, fluctuations in parking duration may be caused by external factors, such as a varying meeting schedule or after-work commitments [10]. The amplitude of these fluctuations depends on the individual EV user and their typical mobility habits [9]. Furthermore, this individuality also extends to a users risk preference, e.g., some users may be willing to sacrifice a fully charged EV for a flexible SC schedule to maximize profit [12].

As a result, any prediction of a user's mobility behavior must quantify this uncertainty, account for a user's individual risk preferences, and integrate this information into the SC algorithm. However, a general uncertainty quantification with a probabilistic prediction may not be sufficient to enable SC acceptance. Not only does a general probabilistic prediction ignore a user's individual risk preferences, such predictions also consider all uncertainty regions equal. However, for a user-centric SC application, the uncertainty resulting in undercharging, i.e. the EV leaves earlier than expected, is more important than the uncertainty that results in a later than expected departure time. Therefore, a customized uncertainty quantification that specifically quantifies uncertainty for SC applications is required. Furthermore, this uncertainty quantification must account for known and frequently visited locations and allow for different levels of uncertainty in these locations. With such a customized uncertainty quantification and location information, stochastic SC algorithms can be applied to improve SC performance [13]–[15].

Thus, the present paper extends our previous work [1] and presents a methodology for customizing the uncertainty quantification of a EV user's predicted mobility requirements specifically for SC applications. More precisely, the present paper focuses on quantifying the uncertainty for predictions of the time a user spends in a given location, i.e. the *parking duration* [15]. Our methodology first consists of data preprocessing to derive known and frequent locations for EV charging and to obtain two prediction labels: parking duration and departure time. We then create a general quantification of the uncertainty associated with these labels through probabilistic predictions. Given these probabilistic predictions, we perform a further customized uncertainty quantification to determine critical errors resulting in undercharging and non-critical errors. Furthermore, we define quantile-based security levels that can be used to minimize the probability of the EV being undercharged, given a user's individual risk preferences. We evaluate our methodology with four different probabilistic prediction models on an openly available semi-synthetic benchmark data set with reduced uncertainty previously presented in [1] and a real-world mobility data set from a single user.

The rest of the present paper is structured as follows. Section II considers existing literature and highlights the identified research gap. In Section III, we present our methodology for customizing uncertainty quantification in parking duration predictions. We discuss the case study used in Section IV before reporting all results in Section V. We analyze and discuss these results in Section VI, before concluding and providing an overview of future work in Section VII.

II. Related Work

As observed in our previous conference paper [1], few researchers have focused on parking duration prediction for a single user [16]. Instead, most research has forecast the demand of an EV either at a charging station, parking lot, or for a fleet of vehicles [27]–[31]. Furthermore, when considering probabilistic predictions, almost all work focuses on electric vehicle charging demand [20]–[26], and does not consider the associated user-specific parking duration.

Considering arrival and departure time, machine learning methods are used to predict these labels in [15]. However, the paper's main focus is on the effects of this prediction on the scheduling and not on the accuracy of the prediction itself. Furthermore, [15] focuses on deterministic departure time predictions for a single location (a workplace) and a fleet of vehicles without quantifying the associated uncertainty. Uncertainty in parking duration and energy demand is considered in [16] via quantile predictions for both quantities. These predictions, however, are only performed for a single location, i.e., the home location [16]. Furthermore, the quantile predictions do not customize the uncertainty quantification for SC applications. The first daily departure time is predicted in [17], and whilst prediction intervals based on an assumed Gaussian distribution of the errors are created, the uncertainty associated with these intervals is not specifically quantified for SC applications. Mobility prediction for many vehicles is considered in [18] to analyze effects on a distribution grid, however, only deterministic predictions are considered. A review of scheduling, forecasting, and clustering strategies for EV charging is provided in [4] focusing on typical scheduling problems and coordinating the charging of multiple vehicles. Whilst probabilistic methods are discussed in [4], they again focus on EV charging demand or EV charging scheduling and do not consider the individual user's parking duration. Further deterministic mobility predictions are considered for a fleet of vehicles in [19] and in the form of next-place prediction in [32]–[34].

As shown in Table I, none of the above papers specifically quantifies the uncertainty in parking duration predictions for user-centric SC applications. Most papers are deterministic and only focus on a single location, or the locations are pre-labeled and not provided as GPS coordinates. When uncertainty is included, it is limited to a single location, only used to compare different predictions, and not customized specifically for use in SC applications. Therefore, we identify a clear requirement for customized uncertainty quantification of parking duration predictions designed specifically for SC applications.

III. Methodology

Our methodology to customize uncertainty quantification of probabilistic parking duration predictions specifically for SC

TABLE I: Overview of related work that considers predicting a user's mobility behavior. None of the identified papers customize the uncertainty quantification of parking duration predictions specifically for SC applications.

Paper	Quantity Predicted	Individual User	Multiple Locations	Probabilistic	Customized Uncertainty Quantification
[16]	Parking Duration & Energy Demand	✓	✗	✓	✗
[15]	Parking Duration	✓	✗	✗	✗
[17]	First Daily Departure Time	✓	✗	✓	✗
[4]	Energy Demand & Schedule	✗	✓	✓	✗
[18]	Energy Demand	✗	✓	✗	✗
[19]	Energy Demand	✗	✗	✗	✗
[20]–[26]	Energy Demand	✗	✗	✓	✗
[27]–[31]	Energy Demand	✗	✗	✗	✗
[32]–[34]	Next-Place	✗	✓	✗	✗

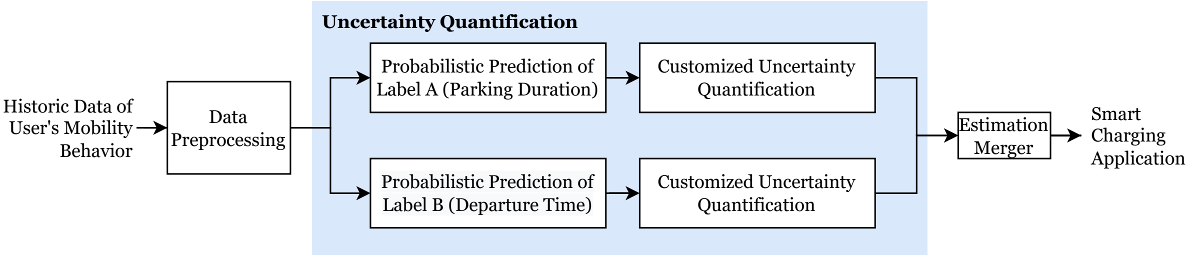

Uncertainty Quantification
Historic Data of User's Mobility Behavior
Data Preprocessing
Probabilistic Prediction of Label A (Parking Duration)
Customized Uncertainty Quantification
Probabilistic Prediction of Label B (Departure Time)
Customized Uncertainty Quantification
Estimation Merger
Smart Charging Application

Fig. 1: Overview of our methodology with the focus of the present paper highlighted in blue. First, we preprocess data to derive known and commonly visited locations and create two prediction labels, parking duration (Label A) and departure time (Label B). Second, we generally quantify the uncertainty associated with parking duration predictions via a probabilistic prediction. Third, we perform a customized uncertainty quantification designed for smart charging applications. Finally, the predictions for both labels can be merged, and this information integrated into a stochastic SC application.

TABLE II: An overview of the hyperparameters for the data preprocessing.

Parameter	Value
Minimum Parking Time	2 h
Maximum Parking Time	24 h
Cluster Density Parameter (DBSCAN)	100 m
Minimum Number of Data Points per Cluster (DBSCAN)	5
Maximum Cluster Distance for Joining Clusters	500 m
Neighborhood Radius to Assign Noise	300 m

applications is shown in Figure 1. The first step is data preprocessing which involves cleaning the data, performing spatial clustering to determine key locations, and data engineering to calculate the prediction labels. The second step creates a general quantification of the uncertainty through probabilistic predictions. Then, in the third step, we customize this uncertainty quantification specifically for SC applications. The final step, which is not considered in the present paper, is merging probabilistic predictions for both labels and integrating this uncertainty into the SC application. In this section, we describe each of the first three steps in detail.[1]

A. Data Preprocessing

The data preprocessing includes three steps: data cleaning, spatial clustering, and data engineering. In the following, we describe these steps and provide an overview of the associated hyperparameters in Table II.

Data Cleaning: The first step in the preprocessing chain is data cleaning. The data cleaning initially involves removing all trips with measurement errors, for example, trips with invalid GPS locations or corrupt time stamps. After removing invalid trips, we calculate the parking duration, i.e. how long the vehicle is stationary before the next trip begins. We then remove all trips shorter than $15\,\mathrm{s}$, since we assume these to be measurement errors and unrealistic trip times. Finally, we aim to focus on parking durations relevant for SC applications. For a parking time of less than $2\,\mathrm{h}$ there is not enough flexibility to enable SC. On the other hand, if the duration exceeds $24\,\mathrm{h}$ there is too much flexibility meaning SC is trivial. Therefore, we filter the data to only include parking durations between $2\,\mathrm{h}$-$24\,\mathrm{h}$.

Spatial Clustering: Given clean data, the next aspect of preprocessing is spatial clustering. Spatial clustering is necessary to determine key locations and account for small fluctuations in GPS coordinates. These fluctuations can occur when the destination is the same, but the exact parking location is slightly different, i.e. a different parking spot at the same supermarket. The spatial clustering consists of two steps. In the first step, we apply standard *DBSCAN*[2] [36] to the GPS parking location of all trips. Although this initial clustering generates several suitable clusters, it creates multiple clusters less than $500\,\mathrm{m}$ apart. Therefore in the second spatially clustering step, we join clusters whose centroids are less than this predefined threshold of $500\,\mathrm{m}$ apart. Such clusters count as a single location for our charging purpose since a user would either walk between them (without using the EV) or, if

[1] An overview of the notation used in all mathematical equations is provided in the *Notation* section at the beginning of the paper.

[2] In this paper we apply the clustering algorithm with *Scikit-Learn* [35].

TABLE III: Additional features generated to provide useful information for the parking duration prediction.

Feature	Description
Current Location	Cluster label for the current location of the EV given as a number.
Hour of Day	Time since midnight given as a real numeric value, e.g., 09:36 am is encoded as 9.6.
Time Window of Day	One-hot encoded variable to indicate different time windows of a day, i.e. morning (5-9am), noon (9am-1pm), afternoon (1-5pm), evening (5-10pm), and night (10pm-5am).
Day of Week	Encoding of the day of the week (numeric value 0-6) to represent frequently recurring weekly trips.
Month	Month of the year (numeric value 1-12) to represent seasonally varying mobility behavior.
Is Holiday	Boolean value based on a holiday calendar indicating which days are public and/or school holidays.
Last Parking Time at Current Location	How long the user stayed at the given location the last time they visited.

they choose to drive, the energy consumption is negligible. Once these clusters are determined, we consider the *noise* points that do not belong to any location cluster. We assign a location labeled as noise to a given cluster if they are within a predefined neighborhood radius of $300\,\mathrm{m}$. We assume this neighborhood radius is a reasonable distance for users to walk when parking at a known location. Finally, we label the clusters according to the frequency of their occurrence, i.e. the cluster that contains the most data points is "cluster 1", that with the second-most data points "cluster 2", and so on. We also label the data points considered as noise with "-1".

In the present paper, we only consider the eight most frequently visited locations for both the training and evaluation of the parking duration prediction. All trips with unknown end locations, i.e. those trips assigned to the *noise* cluster, are removed. This decision is made because a SC application is not possible if the location, and as a result, the charging infrastructure available is unknown. Therefore we only predict parking duration for commonly visited and known locations.

Data Engineering: We engineer specific features from the end time of each trip which are designed to provide useful information for the parking duration prediction. These additional features are shown and explained in Tab. III.

We also generate labels for the parking duration prediction. We select two separate labels for prediction, i.e.

Label A (Parking Duration): The time delta between arriving at the current location and departing for the next destination.

Label B (Departure Time): The point in time at which the next departure will occur, given the current location.

B. General Uncertainty Quantification

We create a general uncertainty quantification for parking duration predictions through probabilistic forecasts. Before defining such probabilistic forecasts, it is important to understand that deterministic predictions implicitly include uncertainty. Such deterministic predictions are formally defined as conditional expectations given the available information [37], i.e.,

$$\hat{y} = \mathbb{E}[Y \mid g, X, \hat{\Theta}], \tag{1}$$

where $\hat{y}$ is the predicted expected value, Y a random variable modeling the parking duration y, g an arbitrary prediction model[3] with estimated parameters $\hat{\Theta}$, and X the exogenous variables used in the prediction. However, whilst this prediction is an expected value that implicitly indicates the presence of variance and uncertainty, deterministic predictions fail to quantify this uncertainty.

Therefore, probabilistic predictions actively quantify the underlying uncertainty. For probabilistic predictions, we no longer consider y as a realization of the random variable $Y \sim f_y$, with the Probability Density Function (PDF) f_y, and the Cumulative Distribution Function (CDF) F_y.[4] Instead, we predict the entire PDF f_y or a subset of information that conveys the uncertainty contained in f_y [37].

Since the information contained in a PDF or CDF is difficult to interpret without expert knowledge [38] and also difficult to integrate into SC algorithms, we create probabilistic predictions in the form of quantile predictions and prediction intervals. We describe both of these prediction forms in the following.

Quantile Prediction: A quantile prediction $\hat{y}^{(\alpha)}$, with nominal level α, is a point prediction with the probability α that the observation y is smaller than the quantile prediction $\hat{y}^{(\alpha)}$ [37], i.e.

$$\mathbb{P}[y \leq \hat{y}^{(\alpha)} \mid g, X, \hat{\Theta}] = \alpha. \tag{2}$$

For example, with $\alpha = 0.5$, the probability of the observation being smaller than the quantile prediction is 50%, equivalent to the median prediction. Whilst the quantile prediction is a point prediction, we can predict multiple quantiles and later combine these predictions to convey information on the resulting PDF or CDF.

Prediction Interval: A prediction interval $\hat{I}^{(\beta)}$, with nominal coverage rate $1-\beta$, is a range of potential values with the probability $1-\beta$ of the observation y being contained in this range [37], i.e.

$$\mathbb{P}[y \in \hat{I}^{(\beta)} \mid g, X, \hat{\Theta}] = 1 - \beta. \tag{3}$$

Usually, these prediction intervals are formed by considering the range between two quantile predictions, i.e.

$$\hat{I}^{(\beta)} = [\hat{y}^{(\underline{\alpha})}, \hat{y}^{(\bar{\alpha})}], \text{ where } \bar{\alpha} - \underline{\alpha} = 1 - \beta, \tag{4}$$

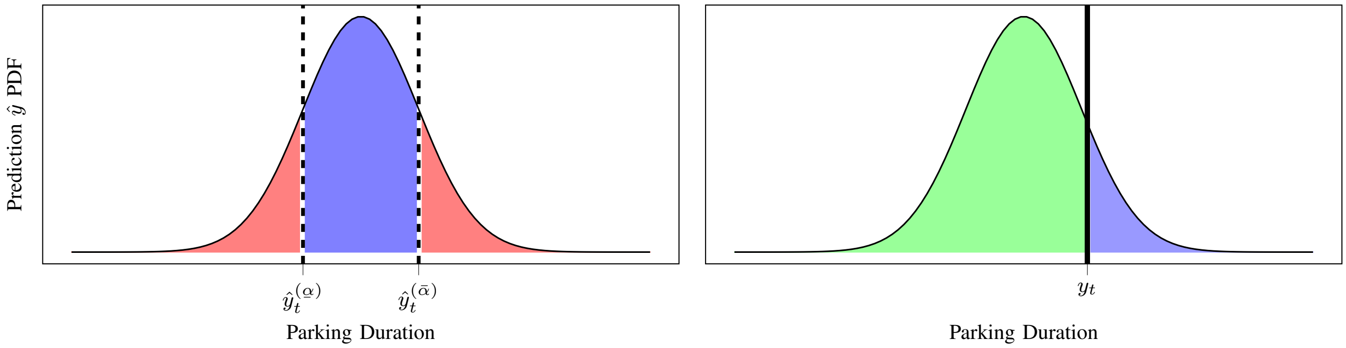
where $\underline{\alpha}$ is the lower quantile, and $\bar{\alpha}$ the upper quantile. To ensure the prediction interval is centered on the PDF, we select symmetrical quantiles around the median, i.e.,

$$\underline{\alpha} = 1 - \bar{\alpha} = \beta/2. \tag{5}$$

Ideal prediction intervals include $(1-\beta)\%$ of the observations.

[3] To clarify, the prediction model g can, theoretically, be deterministic or probabilistic in nature. The expectation in Equation (1) is dependent on g, X, and $\hat{\Theta}$, and thus flexible. For example, with a deterministic g the expectation is formed implicitly through the training process or loss function. On the other hand, with a probabilistic g, the expectation would be performed on the resulting probabilistic prediction to determine, for example, the mean or median.

[4] Note, we use f_y and F_y to define the PDF and CDF of y, respectively. These definitions should not be confused with the above-mentioned prediction model g.



(a) Error Outside the Prediction Interval

(b) Critical and Non-Critical Errors

Fig. 2: A schematic representation of two options to customize the uncertainty quantification of parking duration predictions specifically for SC applications. The error outside the prediction interval shown in red in (a), assumes that the SC application is only at a disadvantage if the EV departs at a time outside the given prediction interval and thus only penalizes observations outside this interval (see Equation (6)). Critical and non-critical error decomposition shown in (b), makes a further distinction between predictions that overestimate the parking duration resulting in the EV leaving earlier than expected and possibly undercharged (critical errors, shown in blue) and predictions that underestimate the parking duration (non-critical errors, shown in green). This error decomposition is introduced in Equation (9).

C. Customized Uncertainty Quantification

Whilst the above-introduced probabilistic predictions quantify the uncertainty of parking duration predictions, they fail to account for regions of uncertainty that may be critical for user-centric SC applications. Therefore, we now introduce three options to customize the uncertainty quantification specifically for SC applications, namely the *error outside the prediction interval*, an *error decomposition*, and *security levels*. We describe each of these options in the following.

Error Outside the Prediction Interval: The first option to customize uncertainty quantification is by comparing the error outside of the prediction interval and the width of that prediction interval. The idea behind this quantification is shown in Figure 2 (a). More precisely, we define the error outside of the prediction interval $\mathcal{E}_{t,\mathrm{PI}}$ as

$$\mathcal{E}_{t,\mathrm{PI}} = \begin{cases} \hat{y}_t^{(\alpha)} - y_t, & \text{if } y_t < \hat{y}_t^{(\alpha)}, \\ 0, & \text{if } \hat{y}_t^{(\alpha)} \leq y_t \leq \hat{y}_t^{(\bar{\alpha})}, \\ y_t - \hat{y}_t^{(\bar{\alpha})}, & \text{if } \hat{y}_t^{(\bar{\alpha})} < y_t, \end{cases} \tag{6}$$

for trip t, and the upper and lower quantile predictions for that trip $\hat{y}_t^{(\bar{\alpha})}$ and $\hat{y}_t^{(\alpha)}$, respectively. This quantification assumes that the only errors relevant for SC applications are those outside a given prediction interval, i.e., the red sections in Figure 2 (a). Therefore, the SC application is only at a disadvantage if the EV departs at a time that is not included in the predicted interval. The $\mathcal{E}_{t,\mathrm{PI}}$ is particularly useful when combined with the average width $\mathcal{W}$ of the prediction interval, defined as

$$\mathcal{W} = \frac{1}{N} \sum_{t=1}^{N} (\hat{y}_t^{(\bar{\alpha})} - \hat{y}_t^{(\alpha)}), \tag{7}$$

for N considered trips. By jointly considering $\mathcal{E}_{\mathrm{PI}}$ and the width of the prediction interval, a SC application can manage the trade-off between a range of possible parking durations and the possible error.

Error Decomposition: Whilst the combination of the error outside the prediction interval and the width of the prediction interval is useful for SC applications, it assumes that errors on both sides of the prediction interval are equal. However, in the case of a user-centric SC application, predictions that overestimate the parking duration, i.e., they predict the EV will depart later than it actually does are far more problematic. Such predictions could lead to an incomplete charging cycle and an EV that cannot reach its destination without additional charging stops.

Therefore, we define the critical and non-critical errors for a user-centric SC application, shown in Figure 2 (b). Beginning with the deterministic case, we consider the total mean error for a parking duration prediction as the mean absolute error between the prediction and actual parking duration

$$\mathcal{E}_{\mathrm{Tot}} = \frac{1}{N} \sum_{t=1}^{N} |\hat{y}_t - y_t|, \tag{8}$$

for each considered trip $t = 1, \ldots, N$. With this definition, we define the critical and non-critical components by rewriting Equation (8) as

$$\begin{aligned} \mathcal{E}_{\mathrm{Tot}} &= \frac{1}{N} \sum_{t=1}^{N} \left(\mathbb{1}_{[\hat{y}_t \geq y_t]} \cdot (\hat{y}_t - y_t) + \mathbb{1}_{[\hat{y}_t < y_t]} \cdot (y_t - \hat{y}_t) \right) \\ &= \underbrace{\frac{1}{N} \sum_{t=1}^{N} \mathbb{1}_{[\hat{y}_t \geq y_t]} \cdot (\hat{y}_t - y_t)}_{\mathcal{E}_{\mathrm{c}}} + \underbrace{\frac{1}{N} \sum_{t=1}^{N} \mathbb{1}_{[\hat{y}_t < y_t]} \cdot (y_t - \hat{y}_t)}_{\mathcal{E}_{\mathrm{nc}}} \\ &= \mathcal{E}_{\mathrm{c}} + \mathcal{E}_{\mathrm{nc}}, \end{aligned} \tag{9}$$

where $\mathcal{E}_{\mathrm{c}}$ is the critical error for the parking duration prediction, $\mathcal{E}_{\mathrm{nc}}$ the non-critical error, and $\mathbb{1}$ the indicator function.

With this definition of critical and non-critical errors we can customize the uncertainty quantification to minimize a certain error as shown in Figure 2 (b). Formally, for a given tolerance

$\mathcal{T}$ we can solve the optimization problem

$$\begin{aligned} \underset{\hat{y}_t}{\text{minimize}} \quad & \mathbb{P}[\hat{y}_t > y_t \mid g, X, \hat{\Theta}] \\ \text{subject to} \quad & \mathcal{E}_{\text{nc}} = \frac{1}{N} \sum_{t=1}^{N} \mathbb{1}_{[\hat{y}_t < y_t]} \cdot (y_t - \hat{y}_t) \leq \mathcal{T}, \end{aligned} \tag{10}$$

i.e., we aim to minimize the probability of a critical error occurring whilst ensuring that the total non-critical error is under a given threshold $\mathcal{T}$. This threshold is, however, highly dependent on a user's individual risk preferences, and therefore a method is required to determine appropriate thresholds.

Security Levels: To help identify an appropriate threshold, we define quantile-based *security levels* that can be used to minimize the critical error depending on an individual EV user's risk preferences. Per definition, a quantile prediction $\hat{y}^{(\alpha)}$ ensures that the probability of the observation y being smaller than the quantile prediction is α. For example, a quantile prediction with $\alpha = 0.9$ should be larger than the observation 90% of the time. Regarding parking duration prediction, this concrete example would also result in a critical error 90% of the time. Therefore, there is a clear mathematical relationship between quantile predictions and the chance of a critical error. To take advantage of this relationship, we define security levels (SL) at level η,

$$\eta = 100 \cdot (1 - \alpha), \tag{11}$$

which minimize the critical error $\mathcal{E}_{\text{c}}$ with increasing η. With this definition, a security level of η should guarantee that for $\eta\%$ of the observations, only non-critical errors occur. Whilst security levels do not determine a user-specific threshold $\mathcal{T}$, they provide a general starting point that can be used to determine approximate thresholds given a user's risk preferences.

IV. Case Study

We evaluate our methodology for customizing the uncertainty quantification of parking duration predictions specifically for SC applications on two data sets and with four probabilistic prediction models. In this section, we introduce these data sets and prediction models before explaining the evaluation metrics applied.

A. Data

To evaluate the proposed approach, we consider two data sets: an openly available semi-synthetic data set with reduced uncertainty that we created for [1] and a real data set based on two years of real mobility behavior that contains the full uncertainty of an EV user[5].

Semi-Synthetic Data: We generate a semi-synthetic data set to create data representing a typical and predictable EV user. The semi-synthetic data set aims to replicate real user behavior, excluding unpredictable events that cannot be accounted for in the optimization process; for example, randomly visiting an unknown location. The semi-synthetic data set thus contains reduced uncertainty compared to the real data.

To achieve this goal, the semi-synthetic data set contains eight locations representing the eight most commonly visited places for a real EV user. Furthermore, there is no "noise" in this semi-synthetic data set, as we assume only known locations are visited. To generate the semi-synthetic data set, we take real travel times between locations from a routing service and multiply them with a normally distributed random factor $k \sim \mathcal{N}(1, 0.05)$ to account for stochastic fluctuations in travel times. Given these trip times, we generate semi-synthetic sequences of trips, including time and location scatter with normally distributed offsets, to replicate the temporal and spatial variation in the trips. Furthermore, the trip sequences include recurrences on four levels: daily, weekly, monthly, and seasonally and two random trips per week to the grocery store occurring with a probability of 50% each. As a result, the semi-synthetic data set still includes uncertainty but unexpected events or trips to unknown locations are removed.[6]

Real Data: The real data is recorded from a personal vehicle over two years, from 2018 to 2019. During these two years, the vehicle was only used by a single person and was also the prioritized means of transport during this time frame.[7] The vehicle was equipped with an onboard computer to record trip data and parking duration. This data was then communicated via an IoT system (using the MQTT protocol) to a backend database for storage (see also [6]). The final data set in the database consists of 2906 trips. Each trip includes GPS coordinates (which we use for spatial clustering) and timestamps (which we use to calculate the time-dependent features and labels). It is important to note that this real data set only mirrors the behavior of one individual and is therefore not necessarily representative of other EV users. Furthermore, since this real data is collected from an individual using their EV to fulfill their personal mobility requirements, the data set contains all uncertainty associated with an EV user.

B. Probabilistic Prediction Models

We use probabilistic predictions to generally quantify the uncertainty associated with parking duration predictions. When selecting probabilistic prediction models, we focus on robust models that are computationally inexpensive, openly available, and proven to perform well. Therefore, we exclude complex deep learning-based regression models that rely on extensive automated feature extraction, are computationally expensive to train, and are not openly available, e.g. [39]–[44]. Furthermore, since there is no clear correlation between successive trips we also exclude time-series-based prediction models that consider auto-regressive terms, e.g. [45]–[48].

Based on our selection criteria, we identify four probabilistic prediction models that are shown in Table IV: Bayesian Ridge Regression (BRR), Gaussian Process Regression (GPR), Natural Gradient Boosting (NGBoost), and a quantile regression Neural Network (NN). Additionally, for each of these models,

[5] Due to privacy concerns we cannot release the real data set openly.

[6] The exact algorithm used for trip generation is openly available and described in more detail on GitHub https://github.com/KarlSchwenk/mobility-data-creator.

[7] Note that the single user was formally notified and agreed to the recording and use of their data for scientific purposes, however, they did not agree to this data being openly released.

TABLE IV: Overview of the selected probabilistic prediction models.

Model	Uncertainty	Distribution	Relationship
BRR	Bayesian Prior	Parametric	Linear
GPR	Covariance Function	Parametric	Non-Linear
NGBoost	Gradient Boosting	Parametric	Non-Linear
NN	Quantile Loss	Non-Parametric	Non-Linear

TABLE V: An overview of the used hyperparameters for the selected probabilistic prediction models.

Model	Hyperparameters
BRR	**Prior Distribution:** Spherical Gaussian **Max Number of Iterations:** 300 **Convergence Tolerance:** 0.003 **Regularization Parameters Noise:** $\alpha_1 = \alpha_2 = 0.000006$ **Regularization Parameters Weights:** $\lambda_1 = \lambda_2 = 0.000006$
GPR	**Kernel:** Radial Basis Function Kernel **Kernel Length Scale:** 1.0 **Noise Regularization:** $\alpha = 0.1$ **Number of Optimizer Restarts:** 10
NGBoost	**Base Regressor:** Random Forest **Number of Trees in Random Forest:** 100 **Random Forest Loss Metric:** Mean Squared Error **NGBoost Distribution:** Gaussian distribution **NGBoost Number of Boosting Iterations:** 10000 **NGBoost Number of Early Stopping Rounds:** 10 **NGBoost Learning Rate:** 0.01 **NGBoost Scoring Rule:** Logarithmic Scoring Rule
NN	**Network Type:** Fully-Connected Feed-Forward Neural Network **Number of Hidden Layers:** 2 **Hidden Layer Size:** 100 Neurons, 50 Neurons **Hidden Layer Activation Function:** Rectified Linear Units **Output Activation Function:** Linear **Epochs:** 200 **Loss Metric:** Pinball Loss **Optimizer:** Adam [49]

we consider a location-dependent ensemble similar to our previous work [1]. The following briefly describes the general idea behind these probabilistic prediction methods and how the location ensemble is created. We refer to the existing literature for detailed mathematical descriptions of the applied models and present an overview of the used hyperparameters in Table V.

Bayesian Ridge Regression: The simplest probabilistic prediction model we apply is BRR. BRR is a Bayesian statistics approach to linear regression that incorporates prior distributions over the model parameters to regularize the estimates [50], [51]. Assuming a linear relationship between the input features and the parking duration target, BRR uses a prior distribution over the coefficients to quantify uncertainty. Since this assumed prior is a parametric distribution, BRR is classified as a parametric prediction method. We implement the BRR in *Python* [52] using *Scikit-Learn* [35] and assume the prior distribution to be a spherical Gaussian. For detailed information regarding BRR, we refer to [53], and [54].

Gaussian Process Regression: Another simple probabilistic prediction model is GPR. The GPR is also based on Bayesian statistics, however, instead of assuming a specific distribution for the prior, it assumes a Gaussian process prior [55]. A Gaussian process is a collection of random variables of which any finite number has a joint Gaussian distribution [56]. As a result, GPR calculates the probability distribution over all admissible functions that fit the data and can therefore model complex non-linear relationships. The considered Gaussian process is defined by a mean function and a covariance function, with uncertainty quantified with the covariance function. We implement the GPR in *Python* [52] using *Scikit-Learn* [35] with Gaussian process prior with a constant mean equal to that of the training data, and a radial basis function kernel with length-scale parameter equal to 1. For detailed information regarding GPR, we refer to [56].

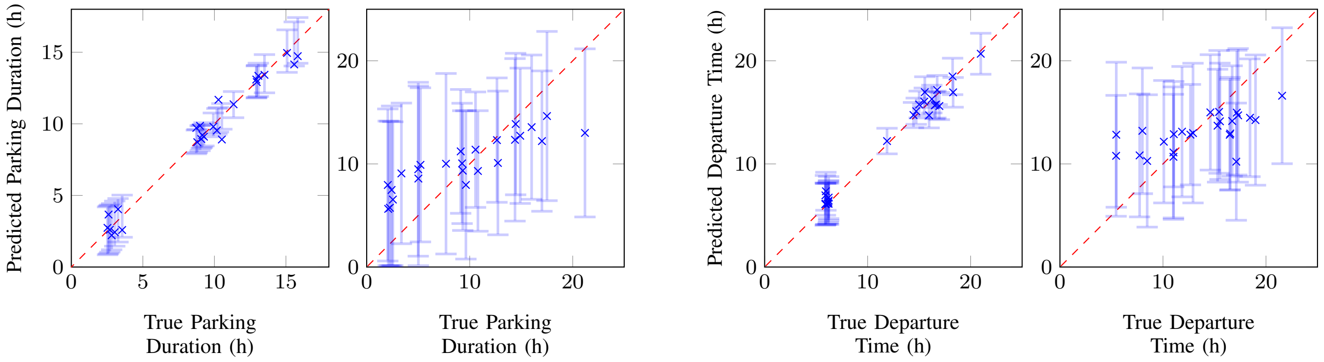
Natural Gradient Boosting: The third probabilistic prediction model is NGBoost, proposed by Duan et al. [57]. NGBoost applies gradient boosting [58] to optimize a probabilistic loss function. More specifically, NGBoost uses multi-parameter boosting and natural gradients to estimate the parameters of an assumed parametric probability distribution. NGBoost is based on an arbitrary deterministic base learner capable of modeling complex non-linear relationships [57]. In the training process, a separate base learner is trained for each parameter of the selected probability distribution using natural gradient boosting to minimize a proper scoring rule [57], such as the logarithmic score or continuous ranked probability score [38]. We implement NGBoost in *Python* [52], with the same random-forest base learner as in our previous work [1] using *Scikit-Learn* [35], and the *NGBoost* [57] python package. We assume a Gaussian distribution [57] and apply the logarithmic proper scoring rule [38] for training. For detailed information regarding NGBoost and gradient boosting in general we refer to [57] and [58].

Quantile Regression Neural Network: The final probabilistic prediction model is a simple feed-forward quantile NN [59]. A quantile NN is trained like any feed-forward NN with gradient back-propagation, however, this training is designed to directly approximate a given target quantile. This approximation is achieved by training a NN with the *pinball* loss, a proper scoring rule whose minimization results in optimal quantile predictions [60]. For a given quantile α, the pinball loss $\mathcal{L}_{\text{pinball},\alpha}$ is defined as

$$\mathcal{L}_{\text{pinball},\alpha} = \begin{cases} \alpha \cdot (\hat{y}^{(\alpha)} - y), & \text{if } \hat{y}^{(\alpha)} \geq y, \\ (1-\alpha) \cdot (y - \hat{y}^{(\alpha)}), & \text{if } \hat{y}^{(\alpha)} < y, \end{cases} \tag{12}$$

where $\hat{y}^{(\alpha)}$ is the prediction for the α quantile, and y the observation [60]. In the present paper, we train multiple NNs to predict multiple quantiles and combine these predicted quantiles by sorting overlapping quantiles to achieve a non-parametric approximation of the full probability distribution. We implement the quantile feed-forward NN with two hidden layers of 100, and 50 neurons respectively. The hidden layers both use the rectified linear units (ReLU) activation function [61], whilst the output layer takes a linear activation function. Similar to [62], we predict 99 quantiles $\alpha \in \{0.01, \dots, 0.99\}$, with individual NNs. The NN is implemented in *Python* [52] using *TensorFlow* [63] with *Keras* [64]. For detailed information on quantile neural networks, we refer to [59], [65], and [66].

Location Ensemble for Each Model: To create a location ensemble, we first separate the training data into the known locations based on spatial clustering to quantify the uncertainty associated with each location. Therefore, the *location ensem-*



(a) Semi-Synthetic, Label A (b) Real, Label A (c) Semi-Synthetic, Label B (d) Real, Label B

Fig. 3: A comparison of the observed and predicted values from the NGBoost model for both Label A and B, on a subset of trips from the test data set. A theoretical perfect deterministic prediction is indicated by the red dotted line, the mean deterministic predictions are blue crosses, and the light blue lines indicate the width of the 90% prediction interval.

ble uses a separate prediction model for each location to generate probabilistic parking duration predictions. Although this allows the models to learn the varying levels of uncertainty at each location, it also leads to higher computational complexity, increasing with the number of known locations. Furthermore, the amount of training data available decreases when only a single location is considered, which may lead to an inaccurate representation of the uncertainty at this location. For each of the four probabilistic prediction models we create a location ensemble, resulting in the BRR Ens, GPR Ens, NGBoost Ens, and NN Ens models.

C. Evaluation Metrics

As evaluation metrics, we first consider a qualitative evaluation of the general uncertainty quantification by visualizing prediction intervals. To further evaluate the general uncertainty quantification, we compare the predicted CDF $\hat{F}(\hat{y})$, with the observed empirical CDF $F(y)$. To this means, we integrate over the absolute difference between the two CDFs, i.e.

$$\mathcal{E}_{\text{INT}} = \int_{2}^{24} |\hat{F}(z) - F(z)| dz, \tag{13}$$

where the integral between 2 h and 24 h is due to the filtered considered parking duration. In this case, a perfectly predicted CDF identical to the empirical CDF would result in an $\mathcal{E}_{\text{INT}}$ of zero, whilst the theoretical maximum $\mathcal{E}_{\text{INT}}$ is 22[8].

To evaluate our methodology for customizing the uncertainty quantification, we consider the metrics defined in Section III-C and the mean error outside the prediction interval for all trips, i.e.,

$$\mathcal{E}_{\text{PI}} = \frac{1}{N} \sum_{t=1}^{N} \mathcal{E}_{t,\text{PI}}. \tag{14}$$

V. Results

In this section, we first compare the general uncertainty quantification from the probabilistic prediction models before reporting the results of our methodology for customizing uncertainty quantification specifically for SC applications.[9]

A. General Uncertainty Quantification

To analyze the general uncertainty quantification from the four probabilistic prediction models we first compare the prediction intervals before reporting differences between the predicted and empirical PDFs.

Prediction Intervals: A comparison of NGBoost predicted and observed values for 25 randomly selected trips from the test data set is shown in Figure 3. The blue crosses are the deterministic predictions obtained as the mean of the probabilistic prediction, and the 90% prediction intervals are shown by light blue bars. For both labels on the semi-synthetic data set, the mean predictions always lie on or close to the diagonal which indicates a perfect theoretical prediction. Furthermore, the prediction intervals for the semi-synthetic data are relatively narrow. Interestingly, prediction intervals for trips occurring in the middle of the considered time range, i.e. around 10 h for Label A and 12 h for Label B, are narrower than those for values at the edge of the considered time range. In contrast to the semi-synthetic data set, the mean predictions for the real data set do not often lie on the desired diagonal. Furthermore, for both Label A and B, the mean predictions overestimate the parking duration for short stops and underestimate this duration for long stops. The prediction intervals on the real data set are also much wider than those from the semi-synthetic data set.

Probability Distribution: To compare the true and predicted probabilistic distribution, we plot the predicted CDF and observed empirical CDF for two locations using NGBoost fin Figure 4. For both data sets, the accuracy of the predicted

[8] Note, this theoretical maximum can only occur in an unrealistic scenario. For example, a predicted CDF predicting the EV will directly depart and thus has a chance of departure of 1 for all parking durations whilst in reality the EV never departed in the considered time period resulting in an empirical CDF with a chance of departure of 0 for all considered parking durations. Since this scenario is unrealistic, we expect $\mathcal{E}_{\text{INT}}$ values lower than this maximum.

[9] Code to replicate the visualizations and error metrics presented in the present paper is available via GitHub: https://github.com/KIT-IAI/Customized-UQ-of-Parking-Duration-Predictions.

TABLE VI: The integral error, $\mathcal{E}_{\text{INT}}$, between the predicted distribution function and the actual observed distribution function for both labels.

		Label A				Label B			
Data	Location	BRR	GPR	NGBoost	NN	BRR	GPR	NGBoost	NN
Semi-Synthetic Data	All Locations	0.9733	2.3763	2.4203	2.6101	2.0008	4.2096	4.2429	4.5634
	Location 1	1.4981	2.1349	2.3448	2.5258	2.7283	3.0856	3.5817	3.5413
	Location 2	0.2521	0.1421	0.1814	0.3618	0.5737	0.5012	0.4837	0.6468
	Location 3	1.8093	0.1752	0.1586	0.1663	0.0388	0.0375	0.0369	0.0896
	Location 4	0.2373	0.2627	0.2133	0.4149	0.2424	0.2378	0.2591	0.3413
	Location 5	1.6834	0.0925	0.0968	0.1167	1.4939	0.0468	0.0439	0.0540
	Location 6	5.6065	0.9957	0.1948	0.5113	7.2874	1.0292	0.2883	0.7282
	Location 7	0.4984	0.5314	0.5371	1.2980	0.4822	0.4953	0.4963	0.7570
	Location 8	0.5027	0.4848	0.5320	0.6274	2.0801	1.8908	1.8918	2.2633
Real Data	All Locations	0.9969	1.2119	1.4666	2.6141	0.9497	0.9855	1.1690	2.1500
	Location 1	1.4773	1.6453	1.8423	3.0827	0.6975	0.7479	0.7965	1.4152
	Location 2	0.6862	0.6840	0.8088	1.5589	0.9972	1.2688	1.2883	1.4998
	Location 3	1.1425	1.4316	1.1238	2.5132	1.3411	1.0643	1.2206	1.0875
	Location 4	0.1626	0.1603	0.1607	0.3090	1.7334	1.5058	2.0576	2.2980
	Location 5	3.8654	3.3779	3.3786	5.3612	3.1125	2.9198	1.2539	2.4804
	Location 6	2.2741	2.9024	3.0737	3.7693	1.6942	1.8172	2.2209	1.7147
	Location 7	0.8766	0.3412	0.2797	0.2812	0.7190	0.6626	0.9439	0.5664
	Location 8	-[a]	-[a]	-[a]	-[a]	-[a]	-[a]	-[a]	-[a]

[a] There was only one trip for Location 8 in the test data set, making the calculation of an observed empirical CDF for this location impossible.

CDF is highly dependent on the location. For example, on the semi-synthetic data set in Location 1 the predicted CDF is underdispersed and struggles to predict trips with either a very short, or very long parking duration. On the other hand, the CDF for Location 2 on the semi-synthetic data set is highly accurate. For the real data, similar results are observed. For this data set, Location 1 results in an accurate CDF, whilst Location 5 is difficult to predict.

To further analyze the deviations in the predicted and true PDF for each location we report the mean integral error, $\mathcal{E}_{\text{INT}}$ for Label A and B in Table VI. With regard to the data sets, the errors are generally lower for the semi-synthetic data than the real data. However, for certain combinations of labels and locations, the errors are lower for the real data set. For both data sets, the location plays a major role. Not only do the errors differ noticeably across the locations, but also the best-performing label changes. For example, Label A is generally more accurate than Label B for the semi-synthetic data, but Label B delivers better results for Locations 3 and 7. Since there was only one observation from Location 8 for the real data, we cannot calculate the empirical CDF for this location.

B. Customized Uncertainty Quantification

The main focus of our evaluation is the customized uncertainty quantification specifically designed for SC applications. Therefore, in this section we first present results for the error outside the prediction interval, before reporting the error decomposition results combined with different security levels.

Error Outside the Prediction Interval: We report the trade-off between $\mathcal{E}_{\text{PI}}$ and $\mathcal{W}$ in Table VII for Label A, and Table VIII for Label B. Comparing these tables, the errors on the semi-synthetic data set are lower than those from the real data, and at the same time the width of the prediction intervals is also smaller. As expected, as the width of the prediction intervals increases, the error outside the prediction intervals decreases.

Considering Label A and the semi-synthetic data set, a minimum $\mathcal{E}_{\text{PI}}$ of $0.03\,\mathrm{h}$ (approximately $1.8\,\mathrm{min}$) is achieved with the BRR location ensemble with a $\mathcal{W}$ of $10.39\,\mathrm{h}$. However, on this semi-synthetic data set, the NGBoost location ensemble performs similarly with a $\mathcal{E}_{\text{PI}}$ of $0.04\,\mathrm{h}$ (approximately $2.4\,\mathrm{min}$) and a lower $\mathcal{W}$ of $2.33\,\mathrm{h}$. On the real data set, the smallest $\mathcal{E}_{\text{PI}}$ of $0.09\,\mathrm{h}$ (approximately $5.4\,\mathrm{min}$) is achieved with the GPR location ensemble, however, the $\mathcal{W}$ is $15.92\,\mathrm{h}$.

For Label B, the lowest $\mathcal{E}_{\text{PI}}$ of $0.07\,\mathrm{h}$ (approximately $4.2\,\mathrm{min}$) on the semi-synthetic data is achieved by NGBoost with a $\mathcal{W}$ of $3.67\,\mathrm{h}$. For the real data, both BRR and the BRR location ensemble achieve the lowest $\mathcal{E}_{\text{PI}}$ of $0.07\,\mathrm{h}$ (approximately $4.2\,\mathrm{min}$), although the $\mathcal{W}$s of $13.92\,\mathrm{h}$ and $13.77\,\mathrm{h}$ respectively are far larger.

Error Decomposition & Security Levels: The error decomposition in $\mathcal{E}_{\text{c}}$ and $\mathcal{E}_{\text{nc}}$ for security levels from 10% to 90% calculated with the NGBoost location ensemble is shown in Figure 5. Although the total error for the real data set is much larger than the semi-synthetic data, a high security level of 90% results in a similar small critical error. Furthermore, for both data sets the minimal total error occurs at a security level between 40% to 60%.

To further analyze this error decomposition, we report the mean critical error $\mathcal{E}_{\text{c}}$, and non-critical error $\mathcal{E}_{\text{nc}}$ for both the semi-synthetic and real data set for Label A, and Label B in Table IX and Table X, respectively. Although the total error is much larger on the real data set, a high security level results in a small critical error for both sets. For example, for Label A we can achieve an average critical error of only $0.03\,\mathrm{h}$ (approximately $1.8\,\mathrm{min}$) on the semi-synthetic data set and $0.1\,\mathrm{h}$ (approximately $6\,\mathrm{min}$) on the real data set, with a security level of 90%. Similarly, for Label B, the same security level can result in an average critical error of $0.01\,\mathrm{h}$ (approximately $0.6\,\mathrm{min}$) on the semi-synthetic data set and $0.09\,\mathrm{h}$ (approximately $5.4\,\mathrm{min}$) on the real data set. Comparing both labels, we observe that the error is not consistently lower for any one label but depends on the considered probabilistic prediction model.

TABLE VII: The mean $\mathcal{E}_{\text{PI}}$ outside of the prediction interval and the average width $\mathcal{W}$ of this prediction interval in hours calculated on the test data for parking duration (Label A). All prediction models and their associated location ensemble (Ens) are compared.

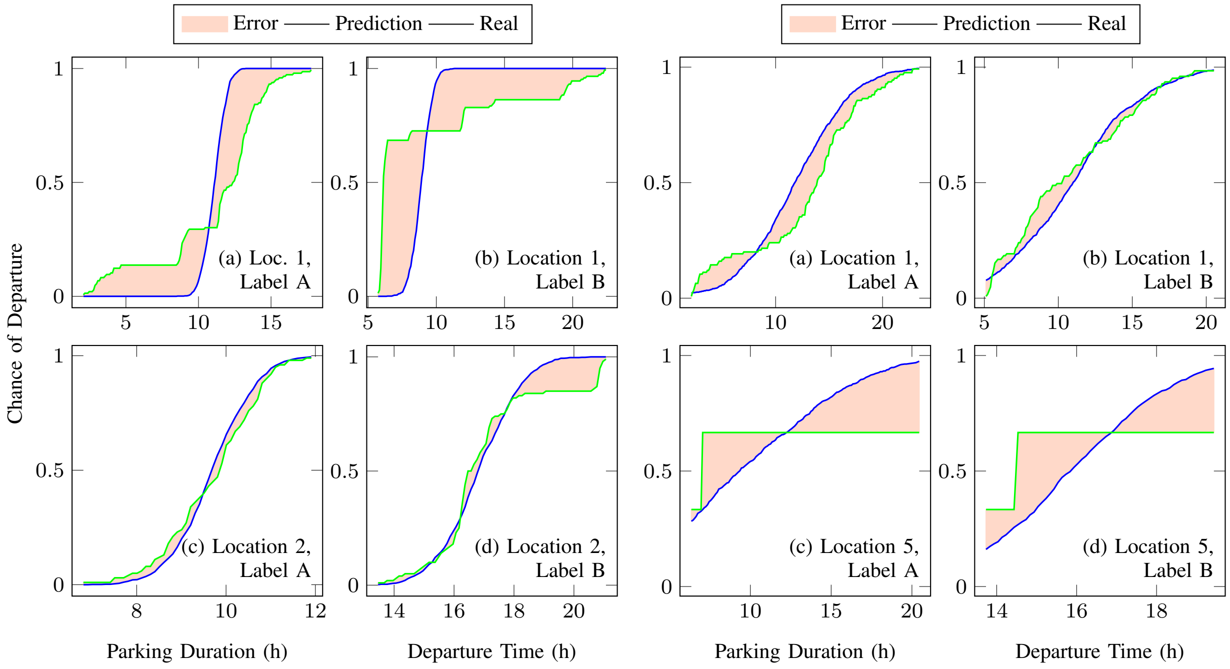
Data	Model	10%-PI		20%-PI		40%-PI		60%-PI		80%-PI		90%-PI	
		$\mathcal{E}_{\text{PI}}$	$\mathcal{W}$	$\mathcal{E}_{\text{PI}}$	$\mathcal{W}$	$\mathcal{E}_{\text{PI}}$	$\mathcal{W}$	$\mathcal{E}_{\text{PI}}$	$\mathcal{W}$	$\mathcal{E}_{\text{PI}}$	$\mathcal{W}$	$\mathcal{E}_{\text{PI}}$	$\mathcal{W}$
Semi-Synthetic Data	BRR	1.64	0.74	1.35	1.48	0.91	2.07	0.58	4.92	0.28	7.50	0.15	9.62
	GPR	0.61	0.19	0.54	0.38	0.41	0.78	030	1.25	0.20	1.91	0.14	2.45
	NGBoost	0.44	0.19	0.38	0.38	0.28	0.78	0.20	1.25	0.12	1.91	0.08	2.45
	NN	0.52	0.11	0.46	0.24	0.36	0.49	0.26	0.81	0.17	1.33	0.11	2.07
	BRR Ens	1.23	0.79	0.97	1.60	0.57	3.31	0.31	5.31	0.11	8.09	0.03	10.39
	GPR Ens	0.57	0.22	0.48	0.44	0.35	0.90	0.24	1.45	0.15	2.21	0.10	2.83
	NGBoost Ens	0.37	0.18	0.32	0.36	0.23	0.74	0.15	1.19	0.08	1.82	0.04	2.33
	NN Ens	0.51	0.11	0.47	0.20	0.39	0.41	0.31	0.67	0.20	1.20	0.13	1.88
Real Data	BRR	3.60	1.31	3.03	2.64	2.03	5.46	1.16	8.76	0.40	13.34	0.14	17.12
	GPR	3.84	1.29	3.28	2.59	2.20	5.36	1.46	8.60	0.67	13.09	0.36	16.80
	NGBoost	3.27	1.12	2.77	2.26	1.88	4.67	1.14	7.50	0.50	11.41	0.22	14.65
	NN	4.35	0.75	3.92	1.60	3.16	3.44	2.38	5.46	1.69	7.75	1.32	9.91
	BRR Ens	3.02	1.24	2.51	2.50	1.63	5.17	0.92	8.29	0.344	12.63	0.14	16.21
	GPR Ens	3.32	1.22	2.79	2.45	1.84	5.07	1.07	8.14	0.36	12.40	0.09	15.92
	NGBoost Ens	2.83	0.92	2.44	1.84	1.76	3.82	1.12	6.13	0.59	9.33	0.34	11.98
	NN Ens	5.02	0.54	4.76	1.08	4.27	2.21	3.65	3.93	2.73	6.21	2.27	7.94

TABLE VIII: The mean $\mathcal{E}_{\text{PI}}$ outside of the prediction interval and the average width $\mathcal{W}$ of this prediction interval in hours calculated on the test data for departure time (Label B). All prediction models and their associated location ensemble (Ens) are compared.

Data	Model	10%-PI		20%-PI		40%-PI		60%-PI		80%-PI		90%-PI	
		$\mathcal{E}_{\text{PI}}$	$\mathcal{W}$	$\mathcal{E}_{\text{PI}}$	$\mathcal{W}$	$\mathcal{E}_{\text{PI}}$	$\mathcal{W}$	$\mathcal{E}_{\text{PI}}$	$\mathcal{W}$	$\mathcal{E}_{\text{PI}}$	$\mathcal{W}$	$\mathcal{E}_{\text{PI}}$	$\mathcal{W}$
Semi-Synthetic Data	BRR	2.55	1.09	2.14	2.20	1.44	4.54	0.82	7.29	0.31	11.11	0.14	14.26
	GPR	0.88	0.26	0.77	0.53	0.58	1.10	0.43	1.77	0.29	2.70	0.22	3.46
	NGBoost	0.59	0.28	0.51	0.56	0.37	1.17	0.25	1.87	0.13	2.85	0.07	3.67
	NN	0.64	0.15	0.57	0.32	0.44	0.65	0.31	1.09	0.21	1.79	0.14	3.07
	BRR Ens	1.83	0.98	1.43	1.98	0.86	4.09	0.52	6.57	0.27	10.00	0.12	12.84
	GPR Ens	0.80	0.32	0.68	0.63	0.50	1.29	0.37	2.08	0.23	3.16	0.15	4.06
	NGBoost Ens	0.67	0.2	0.60	0.40	0.47	0.83	0.36	1.34	0.24	2.03	0.18	2.61
	NN Ens	0.72	0.15	0.64	0.33	0.51	0.69	0.39	1.10	0.27	1.70	0.20	2.73
Real Data	BRR	2.61	1.06	2.16	2.14	1.37	4.44	0.72	7.12	0.25	10.84	0.07	13.92
	GPR	3.15	1.03	2.69	2.08	1.86	4.31	1.10	6.92	0.48	10.53	0.28	13.52
	NGBoost	2.64	0.93	2.24	1.88	1.51	3.89	0.86	6.24	0.29	9.50	0.10	12.19
	NN	3.97	0.62	3.62	1.30	2.92	2.75	2.35	4.19	1.80	5.91	1.44	7.51
	BRR Ens	2.49	1.05	2.04	2.12	1.26	4.39	0.62	7.05	0.19	10.73	0.07	13.77
	GPR Ens	2.69	0.99	2.25	1.99	1.43	4.12	0.74	6.61	0.22	10.07	0.11	12.92
	NGBoost Ens	2.61	0.90	2.21	1.81	1.43	3.74	0.77	6.01	0.27	9.14	0.14	11.74
	NN Ens	3.66	0.64	3.35	1.36	2.77	2.73	2.19	4.15	1.63	6.02	1.40	7.09

TABLE IX: The mean critical error $\mathcal{E}_{\text{c}}$ and non-critical error $\mathcal{E}_{\text{nc}}$ in hours on the test data set for different security levels (SL), when predicting the parking duration (Label A). All prediction models and their associated location ensemble (Ens) are compared.

Data	Model	SL 10		SL 20		SL 30		SL 40		SL 50		SL 60		SL 70		SL 80		SL 90	
		$\mathcal{E}_{\text{c}}$	$\mathcal{E}_{\text{nc}}$	$\mathcal{E}_{\text{c}}$	$\mathcal{E}_{\text{nc}}$	$\mathcal{E}_{\text{c}}$	$\mathcal{E}_{\text{nc}}$	$\mathcal{E}_{\text{c}}$	$\mathcal{E}_{\text{nc}}$	$\mathcal{E}_{\text{c}}$	$\mathcal{E}_{\text{nc}}$	$\mathcal{E}_{\text{c}}$	$\mathcal{E}_{\text{nc}}$	$\mathcal{E}_{\text{c}}$	$\mathcal{E}_{\text{nc}}$	$\mathcal{E}_{\text{c}}$	$\mathcal{E}_{\text{nc}}$	$\mathcal{E}_{\text{c}}$	$\mathcal{E}_{\text{nc}}$
Semi-Synthetic Data	BRR	3.95	0.08	2.85	0.26	2.09	0.44	1.52	0.65	1.05	0.93	0.70	1.31	0.47	1.88	0.31	2.64	0.20	3.80
	GPR	1.02	0.10	0.74	0.15	0.56	0.21	0.43	0.27	0.33	0.37	0.26	0.49	0.21	0.63	0.15	0.82	0.10	1.09
	NGBoost	0.99	0.06	0.70	0.11	0.51	0.15	0.36	0.20	0.25	0.28	0.18	0.40	0.13	0.56	0.10	0.75	0.06	1.04
	NN	0.63	0.12	0.47	0.17	0.38	0.23	0.31	0.27	0.25	0.33	0.19	0.39	0.13	0.48	0.09	0.60	0.04	0.86
	BRR Ens	4.01	0.01	2.67	0.05	1.80	0.18	1.19	0.42	0.76	0.79	0.55	1.38	0.40	2.04	0.26	2.82	0.11	3.87
	GPR Ens	1.18	0.05	0.85	0.10	0.63	0.16	0.47	0.22	0.35	0.32	0.26	0.45	0.19	0.61	0.13	0.83	0.09	1.17
	NGBoost Ens	0.91	0.04	0.64	0.08	0.46	0.13	0.32	0.18	0.21	0.25	0.14	0.36	0.10	0.51	0.06	0.70	0.03	0.98
	NN Ens	0.54	0.13	0.41	0.19	0.34	0.23	0.28	0.27	0.24	0.31	0.20	0.39	0.16	0.46	0.13	0.57	0.06	0.85
Real Data	BRR	6.99	0.27	5.02	0.58	3.75	0.96	2.82	1.45	2.14	2.09	1.58	2.85	1.07	3.74	0.58	4.90	0.13	6.72
	GPR	6.93	0.38	5.04	0.74	3.82	1.13	2.92	1.62	2.23	2.22	1.66	2.94	1.17	3.84	0.72	5.01	0.29	6.79
	NGBoost	6.24	0.33	4.57	0.62	3.46	0.92	2.65	1.33	2.01	1.80	1.45	2.37	0.96	3.09	0.53	4.07	0.17	5.65
	NN	4.08	1.31	3.65	1.52	3.14	1.79	2.68	2.08	2.26	2.47	1.84	2.84	1.37	3.45	0.86	4.20	0.38	5.36
	BRR Ens	5.98	0.24	4.12	0.55	2.95	0.94	2.13	1.46	1.51	2.09	1.05	2.88	0.69	3.85	0.37	5.08	0.10	6.93
	GPR Ens	5.63	0.16	3.83	0.50	2.74	0.94	2.05	1.56	1.58	2.32	1.22	3.19	0.90	4.17	0.57	5.38	0.20	7.14
	NGBoost Ens	4.30	0.40	3.01	0.71	2.24	1.10	1.67	1.51	1.25	2.02	0.93	2.61	0.66	3.34	0.41	4.24	0.20	5.63
	NN Ens	3.22	2.18	2.99	2.47	2.62	2.75	2.36	2.96	2.11	3.18	1.81	3.48	1.53	3.87	1.18	4.59	0.55	5.63



(a) Semi-Synthetic

(b) Real

Fig. 4: The predicted CDF and the observed CDF for the semi-synthetic data set (a) and real data set (b) calculated with the test data for two different locations using the NGBoost model. The error between the two distributions is the highlighted area between the two curves.

TABLE X: The mean critical error $\mathcal{E}_{\text{c}}$ and non-critical error $\mathcal{E}_{\text{nc}}$ in hours on the test data set for different security levels (SC), when predicting the departure time (Label B). All prediction models and their associated location ensemble (Ens) are compared.

Data	Model	SL 10		SL 20		SL 30		SL 40		SL 50		SL 60		SL 70		SL 80		SL 90	
		$\mathcal{E}_{\text{c}}$	$\mathcal{E}_{\text{nc}}$	$\mathcal{E}_{\text{c}}$	$\mathcal{E}_{\text{nc}}$	$\mathcal{E}_{\text{c}}$	$\mathcal{E}_{\text{nc}}$	$\mathcal{E}_{\text{c}}$	$\mathcal{E}_{\text{nc}}$	$\mathcal{E}_{\text{c}}$	$\mathcal{E}_{\text{nc}}$	$\mathcal{E}_{\text{c}}$	$\mathcal{E}_{\text{nc}}$	$\mathcal{E}_{\text{c}}$	$\mathcal{E}_{\text{nc}}$	$\mathcal{E}_{\text{c}}$	$\mathcal{E}_{\text{nc}}$	$\mathcal{E}_{\text{c}}$	$\mathcal{E}_{\text{nc}}$
Semi-Synthetic Data	BRR	6.11	0.30	4.47	0.57	3.34	0.81	2.42	1.06	1.65	1.40	1.07	1.92	0.63	2.65	0.25	3.63	0.01	5.26
	GPR	1.57	0.16	1.18	0.22	0.91	0.29	0.70	0.37	0.54	0.47	0.40	0.60	0.29	0.78	0.21	1.02	0.14	1.41
	NGBoost	1.58	0.05	1.15	0.11	0.86	0.17	0.61	0.23	0.41	0.30	0.28	0.46	0.20	0.68	0.14	0.97	0.08	1.40
	NN	0.84	0.17	0.66	0.22	0.52	0.27	0.40	0.33	0.31	0.40	0.24	0.48	0.17	0.57	0.09	0.75	0.04	1.16
	BRR Ens	5.16	0.24	3.67	0.47	2.62	0.66	1.79	0.89	1.11	1.19	0.54	1.61	0.20	2.29	0.05	3.34	0.02	4.90
	GPR Ens	1.61	0.18	1.16	0.26	0.85	0.34	0.60	0.43	0.40	0.54	0.25	0.71	0.16	0.95	0.10	1.29	0.05	1.78
	NGBoost Ens	1.06	0.15	0.77	0.21	0.58	0.28	0.44	0.35	0.33	0.44	0.25	0.56	0.19	0.72	0.14	0.92	0.10	1.22
	NN Ens	0.70	0.21	0.54	0.29	0.43	0.36	0.35	0.44	0.26	0.52	0.21	0.63	0.15	0.77	0.10	0.95	0.06	1.27
Real Data	BRR	5.82	0.13	4.16	0.34	2.09	0.61	2.30	0.97	1.69	1.43	1.19	2.00	0.76	2.71	0.39	3.68	0.11	5.27
	GPR	5.89	0.25	4.32	0.49	3.38	0.85	2.63	1.22	2.00	1.63	1.47	2.14	1.01	2.79	0.60	3.69	0.23	5.13
	NGBoost	5.18	0.13	3.78	0.36	2.90	0.65	2.23	0.99	1.69	1.39	1.25	1.89	0.86	2.50	0.50	3.31	0.16	4.61
	NN	3.60	1.38	3.21	1.58	2.83	1.76	2.42	2.01	2.06	2.23	1.60	2.50	1.16	2.84	0.78	3.33	0.41	4.11
	BRR Ens	5.44	0.08	3.84	0.32	2.83	0.64	2.08	1.03	1.49	1.50	1.01	2.07	0.62	2.82	0.31	3.83	0.11	5.48
	GPR Ens	5.22	0.13	3.75	0.40	2.84	0.73	2.16	1.12	1.61	1.56	1.14	2.08	0.70	2.71	0.34	3.60	0.09	5.07
	NGBoost Ens	4.85	0.16	3.51	0.39	2.71	0.72	2.10	1.08	1.58	1.47	1.12	1.91	0.72	2.47	0.38	3.26	0.12	4.57
	NN Ens	3.28	1.19	2.81	1.45	2.37	1.77	1.90	2.10	1.56	2.39	1.25	2.80	1.00	3.13	0.73	3.53	0.44	4.37

VI. Discussion

In this section, we first discuss the general uncertainty quantification before analyzing our customized uncertainty quantification specifically for SC applications.

General Uncertainty Quantification: The first observation from the general uncertainty quantification is that the increased uncertainty in the real data set is directly visible. When comparing the two data sets, the prediction intervals for the real data set are far wider than those for the semi-synthetic data set for all prediction models and labels.

The second observation is that uncertainty is highly location-dependent. For certain locations, the predicted CDF is similar to the observed empirical CDF, whilst in other locations, this prediction differs noticeably. Furthermore, both Label A and Label B can be beneficial depending on the location. This result is not surprising since a regular parking duration characterizes some locations (e.g., visiting the gym), whilst other locations are characterized by a regular departure time (e.g., leaving work at the end of the working day).

Finally, the general uncertainty quantification emphasizes the need for a customized uncertainty quantification specifically for SC applications. More specifically, although a prediction interval or CDF is useful for visualizing uncertainty, a SC algorithm will have difficulty optimizing a charging schedule

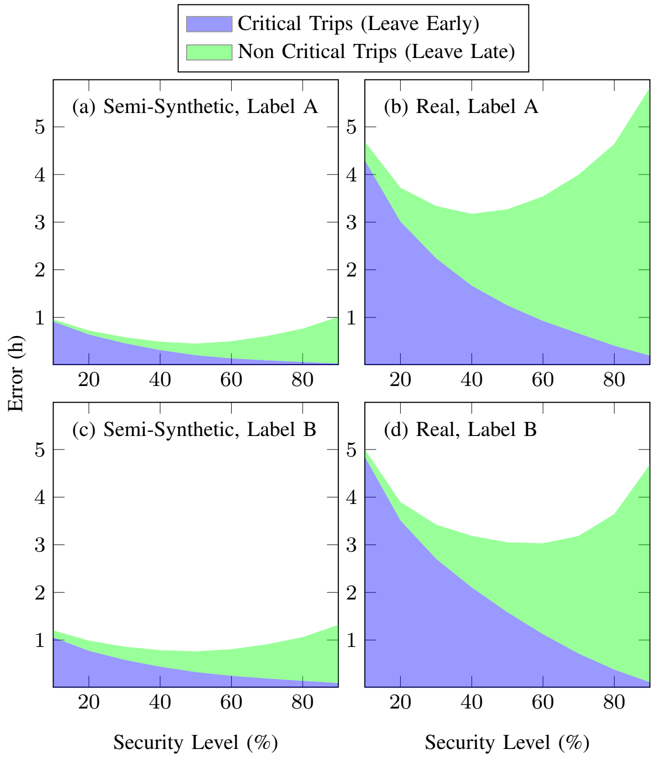


Fig. 5: The total mean error broken down into critical and non-critical error components for different security levels for the NGBoost ensemble calculated on the test data using all locations. The critical error decreases as the security level increases.

based purely on a large range of possible parking durations or an entire CDF.

Customized Uncertainty Quantification: We first observe that quantifying the trade-off between the error outside the prediction interval and the prediction interval width only has limited benefit. To achieve small errors outside the prediction interval, we observe that large prediction intervals are required, which may not be viable in SC algorithms. Furthermore, the error outside the prediction interval does not explicitly quantify critical scenarios that may lead to undercharging.

Therefore, we observe that security levels based on probabilistic predictions combined with an error decomposition provide the most useful quantification for SC applications. This customized quantification method reduces the critical error to acceptable levels, even for real data exhibiting high uncertainty levels. Furthermore, using security levels can account for individual user's risk preferences and can be combined with a SC scheduling to optimize the SC application for that individual user. Given information regarding a user's risk attitude, it may be possible to optimally select a SC schedule that perfectly fits their profile.

However, we currently focus on user-centric SC applications, but the impact on other participants in an IoT smart grid should be considered. For example, an EV charging station owner who aims to optimally schedule charging slots will consider errors resulting in underestimated parking duration critical since the charging slot is not free for an additional EV when expected. Therefore, multiple definitions of critical errors, their associated security levels, and the impacts on an IoT smart grid should be analyzed.

VII. Conclusion

To increase acceptance of Smart Charging (SC) applications, the present paper extends our previous work [1] by introducing a methodology to customize uncertainty quantification of parking duration predictions specifically for SC applications. We generally quantify the uncertainty with probabilistic forecasts before customizing this uncertainty quantification by decomposing critical errors that result in undercharging and non-critical errors. Furthermore, we define quantile-based security levels, which can minimize the probability of an Electric Vehicle (EV) being undercharged, given a user's risk preferences.

Using four probabilistic prediction methods, we evaluate our approach on an openly available semi-synthetic data set and a real data set. We show that uncertainty is highly location-dependent and that a general uncertainty quantification does not provide the specific information required by SC applications. However, our customized uncertainty quantification does provide such information by enabling critical errors to be reduced to acceptable levels for SC algorithms, even when high uncertainty exists in the data.

In light of these findings, probabilistic prediction models that automatically select the optimal label based on location-dependent uncertainty could be considered. Since the present paper focused on user-centric SC applications, future work should consider all participants in an Internet of Things (IoT) smart grid. Specifically, this work should investigate how the definition of critical errors varies for each participant and how these, perhaps contradictory, preferences can be combined to benefit all participants mutually. Furthermore, future work should focus on taking the customized uncertainty quantification presented in this paper and integrating it into stochastic optimization problems, similar to [67], or using it to detect unusual behavior, similar to [68].

Acknowledgment

The authors thank Mercedes-Benz AG, Sindelfingen, Germany for the provided data and allocated resources. This project is funded by the Helmholtz Association's Initiative and Networking Fund through Helmholtz AI, the Helmholtz Association under the Program "Energy System Design".

References

[1] K. Schwenk, K. Phipps, B. Briegel, V. Hagenmeyer, and R. Mikut, "A benchmark for parking duration prediction of electric vehicles for smart charging applications," in *2021 IEEE Symposium Series on Computational Intelligence (SSCI)*, Orlando, FL, USA, Dec. 2021, pp. 1–8.

[2] R. Cornell, "The climate change mitigation potential of electric vehicles as a function of renewable energy," *The International Journal of Climate Change: Impacts and Responses*, vol. 11, no. 1, pp. 15–24, Mar. 2019.

[3] J. García-Villalobos, I. Zamora, J. San Martín, F. Asensio, and V. Aperribay, "Plug-in electric vehicles in electric distribution networks: A review of smart charging approaches," *Renewable and Sustainable Energy Reviews*, vol. 38, pp. 717–731, Oct. 2014.

[4] A. S. Al-Ogaili *et al.*, "Review on scheduling, clustering, and forecasting strategies for controlling electric vehicle charging: Challenges and recommendations," *IEEE Access*, vol. 7, pp. 128 353–128 371, Sept. 2019.

[5] Q. Wang, X. Liu, J. Du, and F. Kong, "Smart charging for electric vehicles: A survey from the algorithmic perspective," *IEEE Communications Surveys Tutorials*, vol. 18, no. 2, pp. 1500–1517, Jan. 2016.

[6] K. Schwenk, T. Harr, R. Großmann, R. R. Appino, V. Hagenmeyer, and R. Mikut, "Data-driven charging strategies for grid-beneficial, customer-oriented and battery-preserving electric mobility," in *Proceedings 29th Workshop Computational Intelligence*, Dortmund, Germany, Nov. 2019, pp. 73–93.

[7] M. Schmidt, P. Staudt, and C. Weinhardt, "Evaluating the importance and impact of user behavior on public destination charging of electric vehicles," *Applied Energy*, vol. 258, p. 114061, Jan. 2020.

[8] K. Schwenk, M. Faix, R. Mikut, V. Hagenmeyer, and R. R. Appino, "On calendar-based scheduling for user-friendly charging of plug-in electric vehicles," in *2019 IEEE 2nd Connected and Automated Vehicles Symposium (CAVS)*, Honolulu, HI, USA, Oct. 2019, pp. 1–5.

[9] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, Feb. 2010.

[10] D. Austin, R. M. Cross, T. Hayes, and J. Kaye, "Regularity and predictability of human mobility in personal space," *PLoS one*, vol. 9, no. 2, p. e90256, Feb. 2014.

[11] Y. Cao, T. Wang, O. Kaiwartya, G. Min, N. Ahmad, and A. H. Abdullah, "An EV charging management system concerning drivers' trip duration and mobility uncertainty," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 4, pp. 596–607, Nov. 2016.

[12] N. Daina, A. Sivakumar, and J. W. Polak, "Electric vehicle charging choices: Modelling and implications for smart charging services," *Transportation Research Part C: Emerging Technologies*, vol. 81, pp. 36–56, Aug. 2017.

[13] K. Schwenk, S. Meisenbacher, B. Briegel, T. Harr, V. Hagenmeyer, and R. Mikut, "Integrating battery aging in the optimization for bidirectional charging of electric vehicles," *IEEE Transaction on Smart Grid*, vol. 12, no. 6, pp. 5135–5145, Nov. 2021.

[14] S. Meisenbacher, K. Schwenk, J. Galenzowski, S. Waczowicz, R. Mikut, and V. Hagenmeyer, "Smart charging of electric vehicles with cloud-based optimization and a lightweight user interface: A real-world application in the energy lab 2.0: Poster," in *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*, Virtual Event, Jun./Jul. 2021, p. 284–285.

[15] O. Frendo, N. Gaertner, and H. Stuckenschmidt, "Improving smart charging prioritization by predicting electric vehicle departure time," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 6646–6653, Oct. 2020.

[16] J. Huber, D. Dann, and C. Weinhardt, "Probabilistic forecasts of time and energy flexibility in battery electric vehicle charging," *Applied Energy*, vol. 262, p. 114525, Mar. 2020.

[17] C. Goebel and M. Voß, "Forecasting driving behavior to enable efficient grid integration of plug-in electric vehicles," in *2012 IEEE Online Conference on Green Communications (GreenCom)*, Piscataway, NJ, USA, Sept. 2012, pp. 74–79.

[18] H. Jahangir *et al.*, "Charging demand of plug-in electric vehicles: Forecasting travel behavior based on a novel rough artificial neural network approach," *Journal of Cleaner Production*, vol. 229, no. 20, pp. 1029–1044, Aug. 2019.

[19] O. Frendo, J. Graf, N. Gaertner, and H. Stuckenschmidt, "Data-driven smart charging for heterogeneous electric vehicle fleets," *Energy and AI*, vol. 1, p. 100007, Aug. 2020.

[20] B. Khaki, Y.-W. Chung, C. Chu, and R. Gadh, "Probabilistic electric vehicle load management in distribution grids," in *2019 IEEE Transportation Electrification Conference and Expo (ITEC)*, Detroit, MI, USA, Jun. 2019, pp. 1–6.

[21] M. Amini and M. P. Moghaddam, "Probabilistic modelling of electric vehicles' parking lots charging demand," in *2013 21st Iranian Conference on Electrical Engineering (ICEE)*, Mashhad, Iran, May 2013, pp. 1–4.

[22] A. Ul-Haq, C. Cecati, and E. El-Saadany, "Probabilistic modeling of electric vehicle charging pattern in a residential distribution network," *Electric Power Systems Research*, vol. 157, pp. 126–133, Apr. 2018.

[23] X. Zhang, K. W. Chan, H. Li, H. Wang, J. Qiu, and G. Wang, "Deep-learning-based probabilistic forecasting of electric vehicle charging load with a novel queuing model," *IEEE Transactions on Cybernetics*, vol. 51, no. 6, pp. 3157–3170, Apr. 2020.

[24] A. T. Thorgeirsson, S. Scheubner, S. Fünfgeld, and F. Gauterin, "Probabilistic prediction of energy demand and driving range for electric vehicles with federated learning," *IEEE Open Journal of Vehicular Technology*, vol. 2, pp. 151–161, Mar. 2021.

[25] L. Buzna *et al.*, "An ensemble methodology for hierarchical probabilistic electric vehicle load forecasting at regular charging stations," *Applied Energy*, vol. 283, p. 116337, Feb. 2021.

[26] L. Petkevicius, S. Saltenis, A. Civilis, and K. Torp, "Probabilistic deep learning for electric-vehicle energy-use prediction," in *17th International Symposium on Spatial and Temporal Databases*, Virtual Event, Aug. 2021, pp. 85–95.

[27] C. Bikcora, L. Verheijen, and S. Weiland, "Density forecasting of daily electricity demand with ARMA-GARCH, CAViaR, and CARE econometric models," *Sustainable Energy, Grids and Networks*, vol. 13, no. 1, pp. 148–156, Mar. 2018.

[28] M. Amini, O. Karabasoglu, M. D. Ilic, K. G. Boroojeni, and S. S. Iyengar, "ARIMA-based demand forecasting method considering probabilistic model of electric vehicles' parking lots," in *2015 IEEE Power & Energy Society General Meeting*, Denver, CO, USA, Jul. 2015, pp. 1–5.

[29] E. S. Xydas, C. E. Marmaras, L. M. Cipcigan, A. S. Hassan, and N. Jenkins, "Forecasting electric vehicle charging demand using support vector machines," in *2013 48th International Universities' Power Engineering Conference (UPEC)*, Dublin, Ireland, Sept. 2013, pp. 1–6.

[30] M. B. Arias and S. Bae, "Electric vehicle charging demand forecasting model based on big data technologies," *Applied Energy*, vol. 183, pp. 327–339, Dec. 2016.

[31] D. Panahi, S. Deilami, and M. Masoum, "Forecasting plug-in electric vehicles load profile using artificial neural networks," Wollongong, NSW, Australia, Sept. 2015, pp. 1–6.

[32] F. Zong, Y. Tian, Y. He, J. Tang, and J. Lv, "Trip destination prediction based on multi-day GPS data," *Physica A: Statistical Mechanics and its Applications*, vol. 515, pp. 258–269, Feb. 2019.

[33] Z. Zhao, H. N. Koutsopoulos, and J. Zhao, "Individual mobility prediction using transit smart card data," *Transportation Research Part C: Emerging Technologies*, vol. 89, pp. 19–34, Apr. 2018.

[34] A. Rossi, G. Barlacchi, M. Bianchini, and B. Lepri, "Modelling taxi drivers' behaviour for the next destination prediction," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 7, pp. 2980–2989, Jul. 2020.

[35] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, Nov. 2011.

[36] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96)*, vol. 96, no. 34, Portland, OR, USA, Aug. 1996.

[37] J. M. Morales, A. J. Conejo, H. Madsen, P. Pinson, and M. Zugno, *Integrating Renewables in Electricity Markets: Operational Problems*. Springer Science & Business Media, 2013, vol. 205.

[38] T. Gneiting and M. Katzfuss, "Probabilistic forecasting," *Annual Review of Statistics and its Application*, vol. 1, pp. 125–151, 2014.

[39] Q. Sun and Z. Ge, "Gated stacked target-related autoencoder: A novel deep feature extraction and layerwise ensemble method for industrial soft sensor application," *IEEE Transactions on Cybernetics*, vol. 52, no. 5, pp. 3457–3468, May 2020.

[40] Q. Zhu, "Latent variable regression for supervised modeling and monitoring," *IEEE/CAA Journal of Automatica Sinica*, vol. 7, no. 3, pp. 800–811, Apr. 2020.

[41] X. Shi, Q. Kang, J. An, and M. Zhou, "Novel L1 regularized extreme learning machine for soft-sensing of an industrial process," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 2, pp. 1009–1017, Mar. 2021.

[42] G. Wang and J. Qiao, "An efficient self-organizing deep fuzzy neural network for nonlinear system modeling," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 7, pp. 2170–2182, May 2021.

[43] K. Jiang, Z. Jiang, Y. Xie, D. Pan, and W. Gui, "Prediction of multiple molten iron quality indices in the blast furnace ironmaking process based on attention-wise deep transfer network," *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–14, Jun. 2022.

[44] G. Wang, J. Bi, Q.-S. Jia, J. Qiao, and L. Wang, "Event-driven model predictive control with deep learning for wastewater treatment process," *IEEE Transactions on Industrial Informatics*, May 2022.

[45] C. Challu, K. G. Olivares, B. N. Oreshkin, F. Garza, M. Mergenthaler, and A. Dubrawski, "N-HITS: Neural hierarchical interpolation for time series forecasting," *arXiv preprint arXiv:2201.12886*, Jan. 2022.

[46] B. Lim, S. Ö. Arık, N. Loeff, and T. Pfister, “Temporal fusion transformers for interpretable multi-horizon time series forecasting,” *International Journal of Forecasting*, vol. 37, no. 4, pp. 1748–1764, Oct./Dec. 2021.

[47] B. N. Oreshkin, D. Carpov, N. Chapados, and Y. Bengio, “N-BEATS: Neural basis expansion analysis for interpretable time series forecasting,” *arXiv preprint arXiv:1905.10437*, May 2019.

[48] D. Salinas, V. Flunkert, J. Gasthaus, and T. Januschowski, “DeepAR: Probabilistic forecasting with autoregressive recurrent networks,” *International Journal of Forecasting*, vol. 36, no. 3, pp. 1181–1191, Jul./Sept. 2020.

[49] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, Jan. 2017.

[50] T. Minka, *Bayesian Linear Regression*. Citeseer, Sept. 2000.

[51] I. Castillo, J. Schmidt-Hieber, and A. Van der Vaart, “Bayesian linear regression with sparse priors,” *The Annals of Statistics*, vol. 43, no. 5, p. 1986–2018, Oct. 2015.

[52] Python Software Foundation, “Python,” 2019. [Online]. Available: https://www.python.org/

[53] M. E. Tipping, “Sparse bayesian learning and the relevance vector machine,” *Journal of Machine Learning Research*, vol. 1, pp. 211–244, Sept. 2001.

[54] D. J. MacKay, “Bayesian interpolation,” *Neural Computation*, vol. 4, no. 3, pp. 415–447, May 1992.

[55] J. Q. Shi and T. Choi, *Gaussian Process Regression Analysis for Functional Data*. CRC Press, 2011.

[56] C. K. Williams and C. E. Rasmussen, *Gaussian Processes for Machine Learning*. MIT Press, 2006, vol. 2, no. 3.

[57] T. Duan *et al.*, “Ngboost: Natural gradient boosting for probabilistic prediction,” in *International Conference on Machine Learning*, Virtual Event, Jul. 2020, pp. 2690–2700.

[58] J. H. Friedman, “Greedy function approximation: A gradient boosting machine,” *Annals of Statistics*, pp. 1189–1232, Oct. 2001.

[59] R. Koenker, V. Chernozhukov, X. He, and L. Peng, *Handbook of Quantile Regression*. CRC press, Oct. 2017.

[60] T. Gneiting, “Quantiles as optimal point forecasts,” *International Journal of Forecasting*, vol. 27, no. 2, pp. 197–207, Apr./Jun. 2011.

[61] A. F. Agarap, “Deep learning using rectified linear units (relu),” *arXiv preprint arXiv:1803.08375*, Mar. 2018.

[62] J. Á. G. Ordiano, L. Gröll, R. Mikut, and V. Hagenmeyer, “Probabilistic energy forecasting using the nearest neighbors quantile filter and quantile regression,” *International Journal of Forecasting*, vol. 36, no. 2, pp. 310–323, Apr./Jun. 2020.

[63] M. Abadi *et al.*, “TensorFlow: Large-scale machine learning on heterogeneous systems,” 2015. [Online]. Available: https://www.tensorflow.org/

[64] F. Chollet, “Keras,” 2015. [Online]. Available: https://github.com/fchollet/keras

[65] T. Gneiting *et al.*, “Model diagnostics and forecast evaluation for quantiles,” *Annual Review of Statistics and Its Application*, vol. 10, Nov. 2023.

[66] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, Nov. 2016.

[67] R. R. Appino, J. Á. G. Ordiano, R. Mikut, V. Hagenmeyer, and T. Faulwasser, “Storage scheduling with stochastic uncertainties: Feasibility and cost of imbalances,” in *2018 Power Systems Computation Conference (PSCC)*, Dublin, Ireland, Jun. 2018, pp. 1–7.

[68] H. Darvishi, D. Ciuonzo, E. R. Eide, and P. S. Rossi, “Sensor-fault detection, isolation and accommodation for digital twins via modular data-driven architecture,” *IEEE Sensors Journal*, vol. 21, no. 4, pp. 4827–4838, Oct. 2020.