TUM Data Innovation Lab

Project Report

Electric Vehicle Charging Pattern Prediction

Simon Klotz, Mariana Martins, Weile Weng

July 18, 2018

Mentor: Dr. Stefan König, sonnen GmbH

Coordinator: Dr. Ricardo Acevedo Cabra

Evaluator: Prof. Dr. Massimo Fornasier

Abstract

For the domestic user with solar energy production capability and an electric vehicle, it's advantageous to maximize the use of solar energy for vehicle charging. Sonnen GmbH wants to accomplish this with its new sonnenCharger. However, implementing such a feature poses a challenge given the lack of information on how much energy is required to charge the battery from its current level to full. This project is set out to create a model that can predict the required energy from charging timeseries. Our dataset comes from Dataport, and consists of electric vehicle charging profiles collected by a charger. After preprocessing, 28287 full charges corresponding to 51 regular EV users were included in the analysis, as well as their extracted features. Several methods such as Ridge regression, XGBoost, and a Conditional Probability model were tested. As a final result an universal approach was derived that allows to predict the required energy as soon as a user signs up for the sonnenCharger, with improving accuracy as more data on a user's behavior is accumulated.

Contents

1 Introduction

The growing adoption of electric vehicles (EVs) around the globe is expected to bring a significant contribution in reducing CO_2 emissions (Quirós-Tortós et al., 2015; Franke and Krems, 2013; Xydas et al., 2016; Robinson et al., 2013). However, it also brings challenges and opportunities to power distribution infrastructures. On one hand, it increases the power demand, and poses potential dangers for peak demand when EVs are charged during current peak hours (Daina et al., 2017; Quirós-Tortós et al., 2015; Robinson et al., 2013). On the other hand, EV batteries can be used as flexible loads providing load balancing services to grids with a large share of renewable energy sources (Daina et al., 2017; Quirós-Tortós et al., 2015).

Smart charging strategies are needed for utilizing renewable energy to the greatest extent while accommodating the current power demand for households as well as the power demand for EV transportation. One important question for the roll-out of smart EV charging services for households equipped with photovoltaic panels is: How to create a charging plan that maximizes the use of solar energy, while still ensuring the vehicle will be charged when the the user needs it?

This question was brought by sonnen GmbH to TUM Data Innovation Lab[1] as the general interest of their student project in the summer semester 2018. Headquartered in Wildpoldsried, Germany, sonnen holds the mission of promoting clean energy by providing intelligent home energy storage systems for private households and small businesses[2]. A predictive model for required energy is needed for the smart charging scheduling function of the sonnenCharger, a smart EV charger launched in February 2018 and introduced to the market in April. The required energy prediction is eventually established as the goal of our project.

This report records our exploratory journey to approach this goal, constituting of acquiring accessible datasets, understanding the nature of the data from interdisciplinary perspectives (engineering, behavioral, business), preprocessing the data and visualizing descriptive statistics, and finally, based on the nature and amount of information embedded in the data, searching for the best predictive model that would accommodate user needs in real business scenarios.

1.1 Overview of sonnen

Sonnen GmbH was founded in 2010 in Wildpolsried, a village in the district of Oberalgäu, Germany, which is marked by its exceptional achievement in renewable energy production and in reducing its carbon footprint. From 2010 to 2015, sonnen focused on developing a solar battery, the sonnenBatterie. Starting from 2016, the company establishes the sonnenCommunity, an energy sharing platform. sonnenCharger, just launched at the beginning of this project, is a smart charger for electric vehicles. Some details of those products relevant to the development of this project are:

sonnenBatterie sonnen's flagship product, a battery for storage of energy produced by home photovoltaic panels, offered in different models with capacity ranging from 2 to 16kWh. The battery features a smart control, which optimizes energy storage based on predicted household usage and electricity production.

[1]https://www.di-lab.tum.de
[2]https://sonnen.de

sonnenCommunity This service allows clients who aren't currently producing enough power (for instance due to bad weather) to receive it from other sonnen households in their region who currently have a surplus. Aside from maximizing the use of clean energy across the community, this also minimizes the purchasing of power from non-solar sources, and consequently the cost.

sonnenCharger The main object of this project is sonnen's newly launched EV charger. It's proposal is to extend the use of clean, free energy from household appliances to electric cars.

1.2 sonnenCharger

While there are available solutions that allow plugging in an electric vehicle directly into a standard residential outlet for charging, they come with drawbacks: since the current and voltage are limited by the capacity of the outlet, full charges often take several hours, even in vehicles with smaller battery capacities. Due to this, a dedicated charging station is the preferred solution for residential households, providing a 230V single-phase, or, if available, 400V three-phase dedicated charging circuit (Morrow et al., 2008).

The sonnenCharger belongs to the latter specification, and beyond providing standard residential EV charger functionality, it also integrates with other sonnen products, providing synergy opportunities to fine-tune functionality of not just the charger, but the home energy environment as a whole, to customer needs.

1.2.1 Charging Modes

The sonnenCharger provides two operation modes, selected by the user through a smartphone app:

Power Charging Mode: This is the default operation mode, where the charger does not impose any restrictions on the charging power aside from those that are safety-related, e.g. cutting off power when the plug is disconnected (for more details on charger operation see Section 1.2.2). As a result, the charging process will be completed in the fastest time possible, as determined by the maximum rated power for the charger and by the vehicle's charge control (including any smart charging features from the EV).

It's important to note that while Power Mode implements the functionality most commonly found in electric vehicle chargers, a fast charge may be undesirable from the user's standpoint. Consider a typical use case where the vehicle is plugged in the evening for overnight charging. A timeline for this use case using Power Mode is illustrated in Figure 2. In this scenario, the charging process will utilize free solar power while it's available from the PV panels or sonnenBatterie, but once these sources are exhausted, it will draw from the power grid at a cost for the user. At a larger scale, electric vehicles charged in this manner place added demand on the power grid, coinciding with peak demand time for domestic use, which can trigger the use of more expensive, and generally less environmentally-friendly power plants at a grid level (Kasten et al., 2016).

Smart Charging Mode: When the user enables Smart Charging Mode, they are prompted through the app for their desired departure time for the vehicle. The sonnen back-end will then determine and send to the charger a charging plan (a time series of charging power

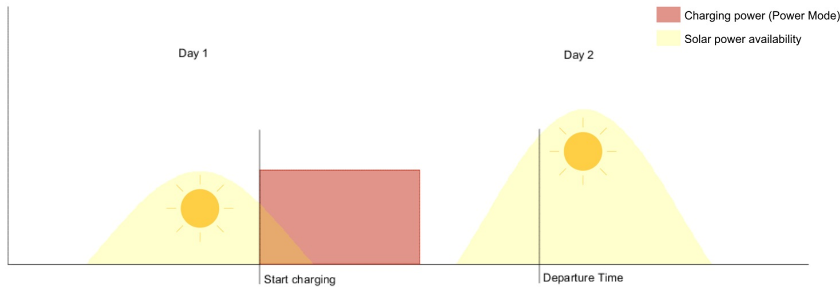


Figure 2: Power drawn vs. solar power availability in an overnight charging event using power mode

limits to be imposed by the charger) that maximizes the use of solar power for the charging process, while still completing the charge before the departure time set by the user. The charger will then limit the charging power according to this profile. Figure 3 illustrates the use of Smart Mode in the previously described scenario.

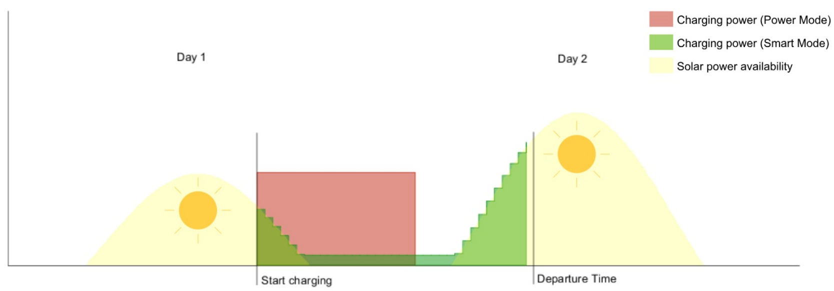


Figure 3: Power drawn vs. solar power availability in an overnight charging event using smart and power modes

1.2.2 Charger-Car Communication and Data Collection

Communication between the charger and vehicle is done through a pilot signal as described by the EN62196-2 standard. Through changing the duty cycle of this signal, the charger communicates to the car the maximum current available for charging (including any additional restrictions from charging profiles in Smart Mode). Meanwhile, the car changes the resistance in the circuit, lowering the signal voltage to specific levels representing the statuses listed below (Toepfer, 2009; Commission et al., 2016):

1. Standby: the charger is not connected to a vehicle;
2. Vehicle detected: the charger is connected to a vehicle, the vehicle does not require charging;
3. Request charging: the charger is connected to a vehicle, the vehicle requires charging;
4. Request charging with ventilation: the charger is connected to a vehicle, the vehicle requires charging, but the battery type is such that the charger should only provide it if it's installed in a well-ventilated location

5. No power and error: Unexpected voltages on the pilot signal indicate a fault in the circuit or a short with a different contact (possibly due to manipulation of the cable).

It's important to note here that while the vehicle can be expected to have extensive instrumentation used by the onboard computer for accurately determining the battery's state of charge, the extremely limited nature of the communication between charger and car means that data available to the charger is limited to it's own instrumentation. The sonnenCharger collects the following data:

- Voltage for each of the three phases
- Current for each of the three phases
- Grid frequency
- Charger temperature

1.2.3 Features of charging process

The power profile of a complete charging event, as measured by the charger, can be seen in Figure 4.

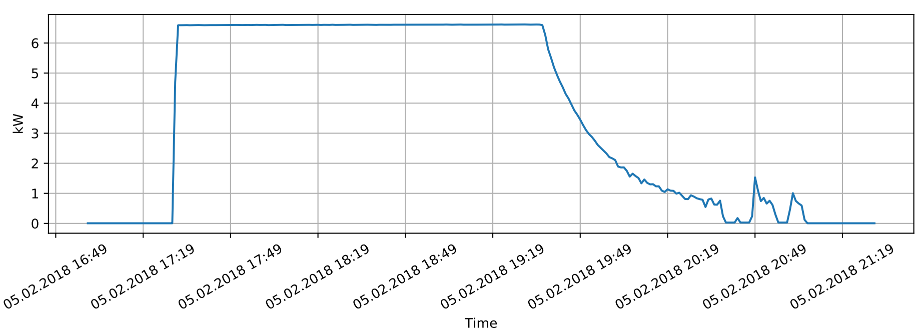
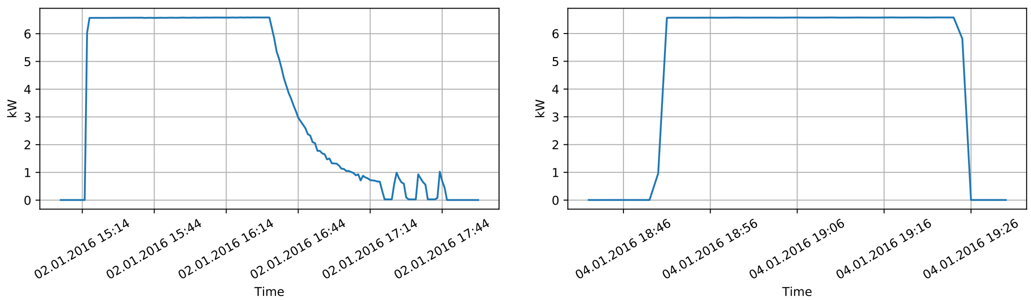


Figure 4: Typical complete charging event, as measured by the charger.

After going over different charging event data collected during development of the sonnenCharger, the following main characteristics were identified:

- Steady state charging power: this is the power drawn during most of the charging process, typically ranging from around 3kW for plug-in hybrid vehicles to 20kW for some pure electric vehicles. This number is determined by the EV's battery management system design, and is constant for a given model.
- Ramp-down: as the battery state of charge approaches 100%, the EV's battery management system will ramp-down the charging rate in order to prevent overcharging, which can impact battery life (Dhameja, 2001) or even cause some types of batteries to ignite (Stuart et al., 2002). Due to this, the presence of this feature in a charging process is an indicator that the battery charged to full, while an abrupt drop in charging power to zero points towards it being interrupted by the user, as illustrated in Figure 5.



(a) Full Charge

(b) Aborted Charge

Figure 5: Comparison of power profiles for a charging event where the EV charged to full capacity and one that was prematurely aborted by the user.

It should be noted that while this behavior seems to be the standard, it is possible that some vehicle models could have different profiles for a full charge, and, if available, the status determined from the pilot line signal described in Section 1.2.2 is a simpler and more reliable indicator of how a charge process was ended.

- Intermittent charging: characterized by short low-power intervals after a full-charge, whose purpose we were unable to determine, though we theorize they might be top-up charges to bring the battery all the way to 100% SoC after the ramp-down reaches the minimum charging power. These are not always present in charging events, and we were unable to identify a correlation of their presence with car models from the development test data.

1.3 Project Objectives

The original DI-LAB project proposal by sonnen was a general one: utilize the data gathered by the charger to optimize the use of solar power in Smart Mode. However, before defining a plan for the charging rates, it's necessary to know how much energy must be charged until full. This is the *required energy*, defined as:

$$\text{Required energy} = \text{Battery capacity} - \text{Current charge}$$

We originally made the assumption that it would be possible to estimate the SoC using the physical quantities measured by the charger and one of the approaches documented in literature. However, upon learning Type 2 chargers such as the sonnenCharger are limited to measuring the AC side of the charging circuit (when most relevant metrics for SoC determination are on the DC side), and about the lack of an interface for obtaining the SoC directly from the vehicle's battery manager, it became clear that the estimation of the required energy alone posed a significant challenge. So, in agreement with sonnen, the project scope was narrowed down to providing a reliable method for this estimation through modeling user charging habits based on the data. With this in mind, we formulated the following questions to be answered by this project:

1. Which are the best methods to predict the required energy?

2. How much data is needed to accurately predict the required energy?

3. How to deal with cases where required energy cannot be reliably predicted?

1.4 Literature Review

To determine the start state of charge (SoC) a priori, the most straightforward way is to collect measurements from sensors directly connected to the battery. If this approach is not available, and no information about SoC is provided by the user, a second information source is the charging profile, i.e. the time series indicating energy inflow. If the time series does not provide any information on the start SoC either, i.e. no matter what the start SoC is, when a user plugs in the charger, the charging profile is indistinguishable except for the amount of time required to charge to full, then the problem is transformed from a problem of engineering science into a problem of behavioral science. A third information source is the charging behavior of the EV user. In the following part, we will review some literature on charging behavior and related results.

Several large scale experiments have been conducted in recent years in different countries to study charging behavior. However, most of them focus on the aggregated impact instead of individual patterns.

Quirós-Tortós et al. (2015) analyzed the data collected from the 'My Electric Avenue' project[3] in the UK which includes more than 68,000 charging events from more than 200 Nissan LEAFs (24 kWh battery capacity, 3.6 kW demand) used by residential UK customers. Each EV was equipped with an inboard monitoring system which registered the start time, end time, initial SoC, and final SoC for each charging event. Probability distribution functions (PDFs) were created as intermediate measures to model the EV demand on the electricity network. Their analysis provides some valuable implications. They found that the charging behaviors showed a more predictable pattern after one week when EV users familiarize with the interaction of the battery level and their driving requirements. Also, charging patterns were different between the first connection and the second connection on the same day. The first EV connection may occur at any time during the day, and a second connection is more likely to occur after midday with higher SoC. During weekends, disconnections before EVs are fully charged are more frequent.

Robinson et al. (2013) quantified the recharging behaviour of a sample of EV users. Their data was records from in-vehicle loggers as part of the SwitchEV trials started in April 2011 in north east of England, including 31,765 EV trips and 70704 EV recharging events (23,805h of recharging), over a 6 month trial period, from 12 private users, 21 organization individuals and 32 organization pool vehicles. It was found that user types and locations affect charging profiles. For instance, private users mainly charged at recharging points at home in the evening, vehicles from organization individuals were primarily recharged upon arrival at work, and organization pool users recharged at work and public recharging points throughout the working day.

Franke and Krems (2013) addressed the individual difference in charging behavior by examining the psychological dynamics underlying EV users. Their data came from 79 EV users in a 6-month field study in the metropolitan area of Berlin. The EV was a converted MINI Cooper. Test drivers had access to a network of 50 public charging points in the metropolitan area as well as private home-based charging points. In addition to logger data, data from interviews, questionnaires, and travel and charging diaries at three stages

[3] see http://myelectricavenue.info

of the trial were collected. They conceptualized user-battery interaction style (UBIS) and developed a measure to assess it. Users with lower UBIS would charge their EV regularly based on contextual triggers regardless of charge level, while users with a higher UBIS only charge when their subjectively preferred charge level is reached. Franke and Krems (2013) showed that the comfortable range (i.e. user's preferred range buffer) together with UBIS (i.e. the tendency of whether a user orients to this level or not), explain when people typically charge their EV. Specifically, users with high scores on these two variables, typically recharge their EV at a lower battery level.

Other research addresses the charging behavior from a theoretical point of view. For instance, considering charging behavior as a consequence of charging choices made for the scheduling of activities and travels, Daina et al. (2017) defines charging choice as the decision made by a driver, at a given charging opportunity, to charge their vehicle to a specific charge level. They assume that individuals make their charging decision once when they arrive at a charging point, having in mind their next travel requirements; and a charging decision is evaluated on utilities of two components, i.e. the utility of charging alternative, and the utility of activity travel timing alternative at a given charging opportunity. The first utility component is affected by the target energy, effective charging time and charging costs; the second utility component is determined by schedule delays and activity participation penalties, as well as total travel time and total travel cost.

Different habits of recharging at a certain time or times of the day would interfere with the amount of time till the end of last charge being a good predictor for the start SoC (or required energy) of current charge. Wang et al. (2011) considered four theoretical recharging scenarios: (1) unconstrained recharging of EVs as soon as a user arrives at home, (2) recharging 3 hours later after arrival, (3) smart charging, (4) smart recharging with demand response. For all users, the source of interference could come from the heterogeneity in charging scenarios among different users, that is, users respond to peak/off-peak hours of the power grid differently. For a single user, such correlation could be interfered when he interchanges charging scenarios constantly.

Accessibility of recharging points outside home could influence the start SoC as well. Weiller (2011) predicted that energy demand from recharging at home will increase by 25.0% to 29.4% if that is the only recharging point available to a driver. If workplace recharging infrastructure is provided, 24.6% and 28.7% of recharging would be expected to occur at work.

In short, to fully capture an individual's charging behaviour, besides records of charging events occurred at a fixed location such as home, additional information is needed, e.g. charging events that occurred outside home with connection to the current charge count for the day (Quirós-Tortós et al., 2015), the user type (e.g. one single user, multiple users) (Robinson et al., 2013), user's UBIS and comfortable range (Franke and Krems, 2013), activity-travel plans, options of other charging points and costs (Daina et al., 2017), home infrastructure of smart charging or preference for charging off-peak hours (Wang et al., 2011), and whether a user only charges at home (Weiller, 2011). Without further information, the predictability of start SoC of full charging events at one location based on the history of charging events at this location, depends strongly on the regularity of an EV users' daily schedule.

As one side note, analysis in Quirós-Tortós et al. (2015) Franke and Krems (2013) only involves one car model. It implies that the heterogeneity of car or battery models could influence the generality of prediction from one user or a group of users to another, besides above discussed differences in charging behaviors.

2 Data Preprocessing

The sonnenCharger was introduced in April 2018 and, prior to that date, only data from sonnen's development tests is available. Therefore, we had to rely on the openly available dataset of Dataport first, and propose the application of our methods to the data from the sonnenCharger as soon as enough data is available.

2.1 Datasets

The Dataport[4] dataset provided by Pecan Street Inc., a non-profit research institute, contains appliance and home-level energy consumption data from volunteers in Austin, Texas. The data can be downloaded as CSV files from the original website or directly from the PostgreSQL database itself. The dataset contains individually measured (disaggregated) power use data of home appliances such as fridges, air-conditioners, and dishwashers from about 722 homes, whereas for 103 of these households the energy consumption for charging an electric vehicle is available.

Two types of EV related datasets are provided by Dataport: (1) *metadata* of the households with EV data available, uniquely identifiable by a `dataid`. It includes the date enrolled or withdrawn (if provided) to the program; (2) *time series* of consumed power (in kW) by EV, for each `dataid`. The resolution of the time series is one minute intervals. Available data starts from the beginning of 2011 up until now.

Each `dataid` with EV data available corresponds to one or multiple EV users of the household. From now on we will use the term *user* synonymously to `dataid` in our report.

2.2 Feature Extraction

From the metadata, a meta table for all users is created by keeping only the relevant information. Additional features of the EV or user behavior are added after preprocessing each time series.

From each time series, a event table is generated where valid charging events are extracted, and related features are identified or calculated. Precisely, the time series is preprocessed in the following steps:

(1) **Extract valid charging event**: a charging event is the part of a time series when the EV power consumption (in kW) stays above a predefined baseline tolerance level, with the start time the first time point when the power exceeds that level, and the end time the first time point when it drops below that level after the start time. A charging event is valid if there exists a time point within the charging period when the power is within a given tolerance of the expected steady value. The `steady state power` a car model uses for charging is determined by its EV battery and onboard charger characteristics. All the valid charging events are extracted from a single time series and each of them is given an unique event number. In our implementation, we define the `baseline tolerance` level to be 0.1kW and the `steady state tolerance` to be 1kW. To simplify the problem, we only show interest in the car model of the `maximum steady power` for a single user, if the user is known to charge different EV models at the same site. For this purpose, at first, we estimate the expected steady state power to be the `maximum power` of the whole time series. However, it is proven to be not a good estimator for non-typical charging profile, where the

[4] see https://dataport.cloud

charging profile oscillates with peaks high above the `steady state power` as depicted in Figure 6. We speculate that the occurrence of this pattern might be due to the car heater being used during the charging process. Since those cases are not rare, we decide to include them. Upon the observation that in those oscillation cases, the peak value does not persist, only the steady value (plus some low-amplitude noise) does, we decide on the following estimation approach: given the time series of a single event, first digitize it into bins with bin size of 0.2kW, define a `bound` variable to be 10 bins, and return the highest value in bins in range of `[bound, mode+bound]` with the `mode` the bin that appear most frequently with the first 10 bins out of consideration.

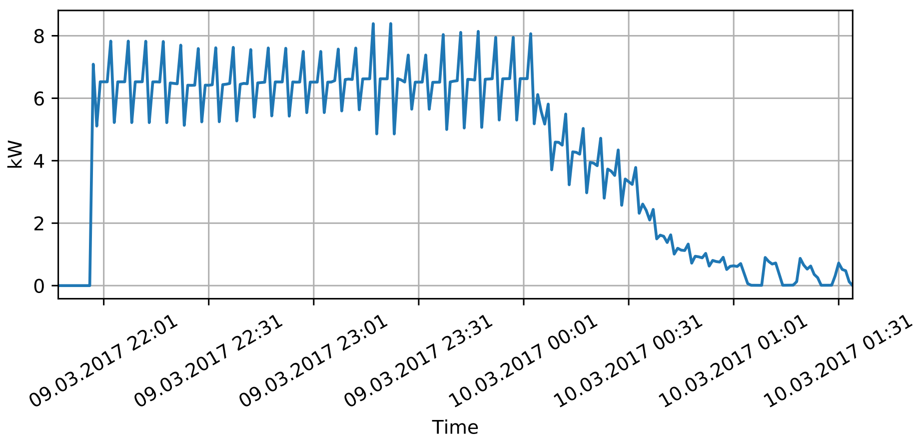


Figure 6: Charging profile with oscillation

(2) **Decompose event into charging stages**: Each valid charging event is further decomposed into at most four stages as depicted in Figure 7.

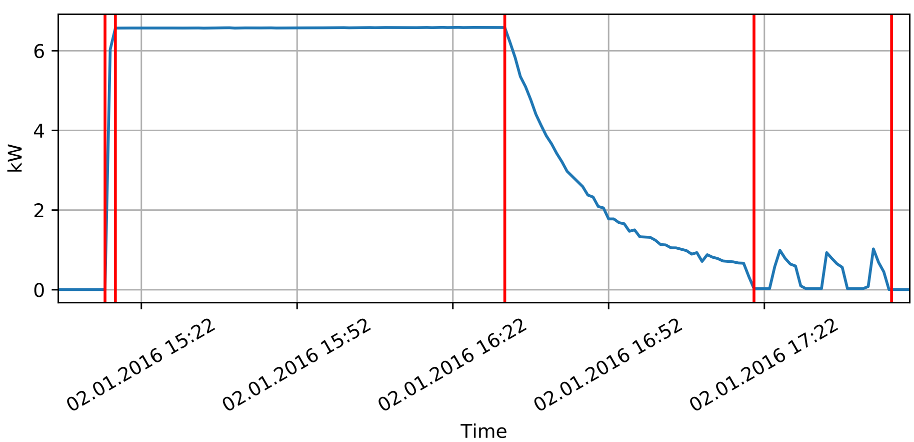


Figure 7: Four-stage-decomposition of the charging profile of a typical charging event

`Stage 1` (or ramp-up stage) is the sub-period from the start time to the first time the power mounts above a lower bound, e.g. 90% of the estimated steady state power. `Stage 2` (or steady-state) is the subsequent sub-period where the power stays close to the expected (`steady state power`) until it's rolling mean drops by a large step away from the `steady state power` by a threshold. We choose rolling mean of window size 5 (minutes) instead of the single input since we don't want to exclude the oscillation cases. We choose both step size and threshold to be 5%. `Stage 3` (or ramp-down stage) is the following sub-period until the power starts to rise with big step from a low level (e.g. 20%), if such case exists. `Stage 4` (or intermittent stage) is the rest sub-period till the end of the whole valid charging event. In case the power does not rise again after the second stage, the rest sub-period is marked as the `stage 3`. The period after the end of such an event and before the begin of the next event is identified as `stage 5` (i.e. the resting stage). The duration, amount of energy charged, and mean energy charged at each stage is calculated[5]. In practice, for some valid charging events, the power ramps up quickly to the steady state such that the first stage is skipped by the decomposition algorithm.

(3) **Identify fully charged events**: A valid charging event is considered fully charged when the `stage 3` lasts more than a certain amount of time (e.g. 2 minutes) or its `stage 4` exists, in accordance with the characteristics mentioned in Section 1.2.3. This step is important because it allows us to evaluate `start SoC` at the same time with (estimated) battery capacity, since `start SoC` can only inferred a posteriori when EV is fully charged.

(4) **Quantify `stage 3`**: Some parameters are developed to describe the dropping curve in `stage 3`, including L1 distance to the mean point, L1 distance to a decreasing linear line, L2 distance to a decreasing linear line, and variance and variation of the power profile in stage 3. We assume the dropping curve provides some information about the EV charging system, possibly allowing inference of the vehicle model. Whether this theory is plausible or not awaits further analysis of a larger data pool with cars where the model is known and consultation with experts. On the other hand, the `stage 4` might also be an information source about the battery model or usage status. Those questions are however not within the scope of our project.

(5) **Estimate other car or user specific features**: The maximum power value in the time series serves as a feature of the EV model. The maximum amount of energy charged in a single charging event is estimated as the battery `capacity` of the EV, since Dataport does not provide such information. Furthermore, features describing a user's charging behavior are generated from the times series, such as the total number of valid charging events, number of active days[6], charging frequency[7], the percentage of fully charged events, average amount of energy charged in a single event, vectors approximating the distribution of the amount of energy charged, of number of valid charging events over days of week and over hours of the day, and `entropy` of normalized distribution vectors that quantify the uncertainty in corresponding charging behavior patterns.

(6) **Extract time-related features**: Timestamps are preserved in local time format. From timestamps that records the begin of a charging event, we extracted the `hour`, `month`, `day`,

[5]Two important features for the time series are generated during the decomposition: the variable `charging timestamps` encodes the critical start point of each stage, with values from {`1, 2, 3, 4`} and `charging status` encodes which stage does a time point belong to, with values from {`0, 1, 2, 3, 4`}, where `0` represents the default resting stage.

[6]It is calculated as the time difference between the start of first charge and the end of last charge in days.

[7]Charging frequency = Number of active days/Number of valid charging events.

day of week, weekend[8], and season. In addition, we also classify hour of start time into clusters of hour with 4-hour interval and 8-hour interval. For each charging event, which connection it counts for the current day or week is also calculated.

We use the total amount of energy of each (full) charging event instead of the energy of single stages in our main analysis. However, in ideal situations when the information provided is sufficient to identify each stage correctly, we would recommend using the energy consumed in stage 2 and its related charging time to characterize the charging behavior pattern. Since the time span of a valid charging event is dominated by the duration of its steady state, where its rolling mean of power stays high compared to other stages. This results in a leading contribution to the total amount of energy charged during that event. In fact, stage 1 and stage 3 are auxiliary stages which are relatively constant for all events, and stage 4 is reducible, while stage 2 is the *purpose*, in other words the *heart*. Therefore, energy charged at stage 2 is more representative than the total energy charged in analyzing the EV charging behavior. The physical properties of EV batteries determine the expected steady charging power, the length and shape of stage 1 and 3, as well as the shape of bumps in stage 4. In contrast, the length of stage 2 (and stage 4, if applicable) is determined by the user who initiates and monitors the charging event.

2.3 Data Cleaning

As mentioned before, we exclude noise events, remove noise from charging profiles, with noise level set to 0.1kW, as well as events that have not reached the expected steady state. Besides, we exclude irregular users, i.e. users with less than 50 fully charged events that have reached the steady state, and users with irregular charging profiles. Figures 8, 9 and 10 respectively show examples of a noisy charging event, a charging event with a noise peak at stage 1, and a charging event with an unusual charging profile.

Essentially, we only include full charges in our training and prediction, since our focus is to predict the required energy for a full charge.

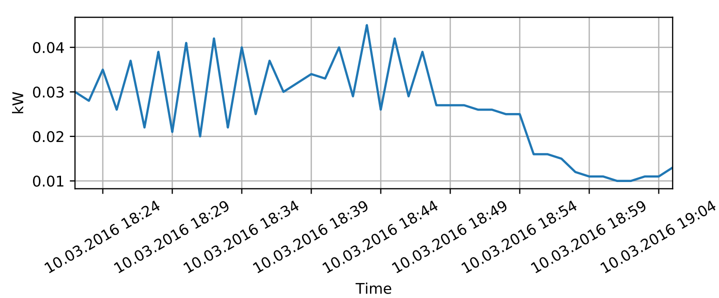


Figure 8: A noisy charging event

Although times series data is available from 2011, we sub-sampled the dataids which are registered before 01.01.2016 and withdrawn after 01.05.2018 and their charging timeseries of this period, since we are interested in charging behavior of recent years, and earlier data is unlikely to reflect the current habits. In total, we collected 33712 valid charging events

[8] A boolean variable.

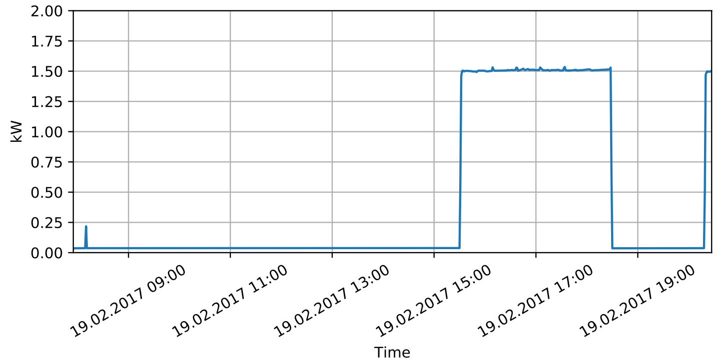


Figure 9: A charging event with a noise peak at stage 1

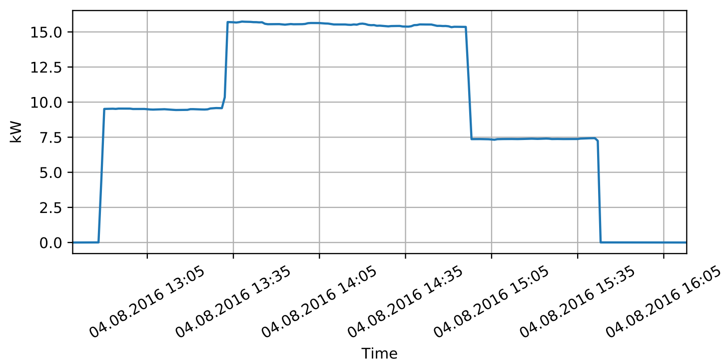


Figure 10: A charging event with unusual profile at stage 1 and stage 3

from 77 dataids, where 27608 of them were charged to full. After cleaning, 51 dataids are identified as regular users, from whom, 27287 of 32328 valid charging events are full-charge, and thus were included in the data analysis. On average, each regular user had 535 full charge events. They were likely to charge to full 84% of the time, with start SoC of 55%, and 28% fully charged events occurred on the weekends. A full charge of a regular user required 6.0037 kWh on average. 50 regular users have EVs with capacity greater than 10kWh, and mostly between 10 and 15kWh (see Figure 15).

2.4 Preliminary Analysis

In this part, we examine three types of questions related to charging behavior, and present the results as preliminary analysis.

(A) **Would the amount of time until the end of last charge be a good predictor for the required energy of current charge?** The answer is no. As observed in Figure 11, the required energy of current charge is relatively invariant to pre-resting minutes, exhibiting a horizontal band of points. A linear regressing including pre-resting minutes as variable would be unsuitable, since the correlation coefficient's value would be steered by sparse points outside the dense range, and the p-value would show false significance. The reason for the false significance is that the range of bounded energy is inherently bounded by battery capacity, while the range of pre-resting minutes is unbounded, which produces very low standard error. One interesting observation is that points are more dense around the range [1200, 1450] , and within this area, a clear line is formed showing negative correlation between required energy and pre-resting minutes. Note that those minutes signify the time point close to 24 hours. And the negative slop of the line is due to that a larger capacity would require more to charge than a small capacity would. If users usually charge at frequency of an integer multiple of 24 hours, then the pre-resting minutes between a current charge to the end of previous charge would correlates negatively to the amount required energy.

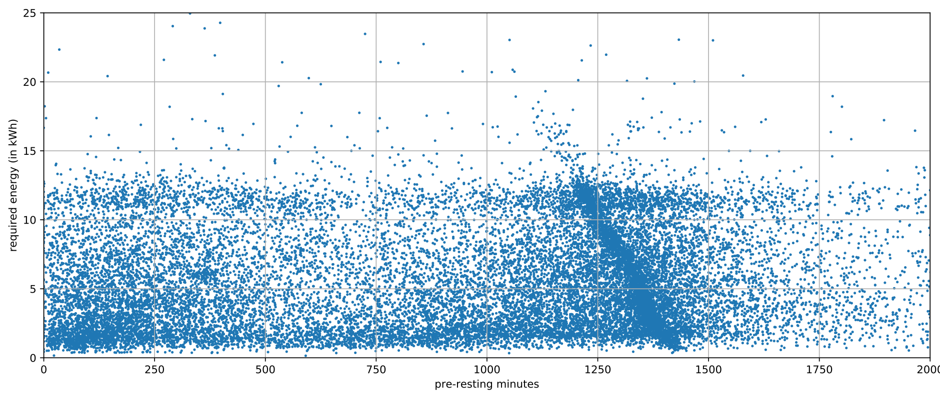


Figure 11: Scatter plot exaiming the correlation between pre-resting minutes and required energy. The plot is cut off for charging events with pre-resting minutes less than 2000, and required energy less than 30 kWh.

(B) **Is there a common pattern in charging behaviors, in terms of start SoC level?** An overview the start SoC distributions of fully charged events for each regular user reveals mixture of three types of users (see Figure 12): (a) users prefer to charge at one level of SoC (low, middle, or high), (b) users have multiple preferences, (c) users have near indifferent preference among most of the start SoC levels.

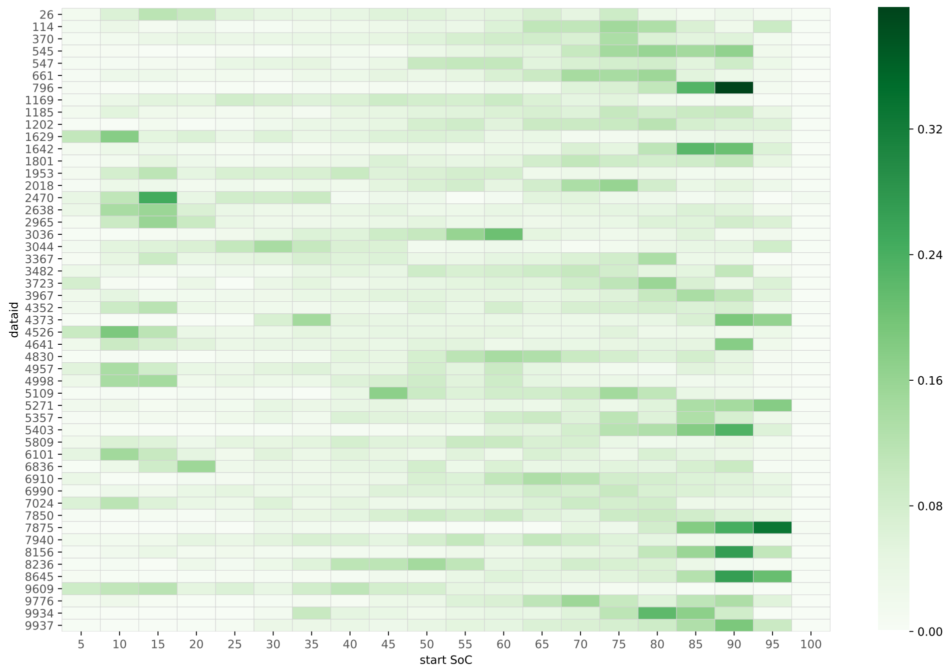


Figure 12: Distribution of start SoCs in fully charged events for regular users. Each row corresponds to a dataid. The darker the color in a cell, the often the user charges at this level. One can observe several types of users: (a) users with preference to charge at one level of start SoC (e.g. low: 1629, 2470 and 4526; middle: 3036, and 5109; or high: 796, 8156, and 9934), (b) users with multiple such preferences (e.g. 547, 661, and 4373), and (c) users with near indifferent preference among most of the start SoC levels (e.g. 1169, 3482, and 7940).

(C) **Does the required energy a user charge on current day depend linearly on the energy charged on previous days? Does the current charged amount depend on the value of previous charges?** To answer these questions, as suggested by Box et al. (2015), we evaluated autocorrelation and partial autocorrelation of required energy in time series (1-day-resolution) for each user, with lag 30 days. We did not find significant correlation: correlations are in general below 0.2 in both cases, which do not differentiate from correlation one would observe from noises. In addition, no systematic pattern were found for all users, besides slightly higher (positive) autocorrelation among same day of week (i.e. in frequency of 7 days) compare to other days (see Figure 13). Same observation apply to autocorrelations with hourly resolution.

Together with the fact that we did not find correlation between pre-resting minutes and required energy, it left only time related categorical features as predictors. This suggests linear regression might not be suitable for the prediction task, and a conditional probabilistic approach might be worth a try.

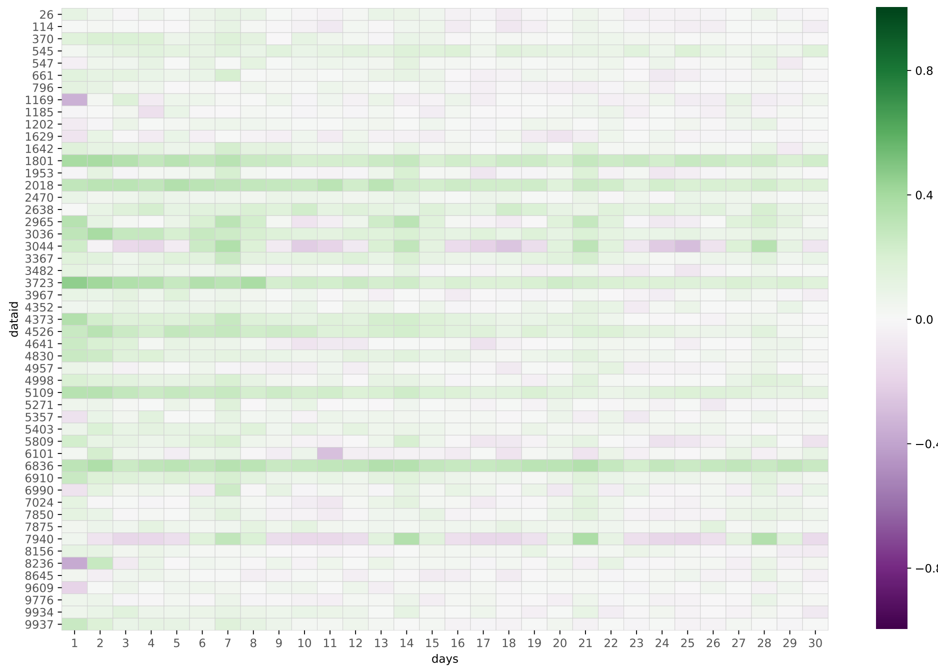


Figure 13: Autocorrelation of required energy for fully charged events of regular users, with lag 30 days. We observe that the correlations are low in general, do not show systematic pattern among all users, and slightly higher autocorrelation among same day of week compared to other days.

3 Methods

This section describes the models used to predict the required energy and how we evaluate them to address our three project objectives.

3.1 Models

As part of our project we employ two different kinds of methods: machine learning models that give point estimation, and conditional probability model (precisely, conditional distribution based model) that would provide a point estimation with definable confidence level. The latter model has access to the conditional distribution, and the freedom to control the range of the confidence interval.

3.1.1 Machine Learning Models

We selected the following common machine learning methods to simply make point predictions of the required energy based on our extracted features.

(1) **Mean Model**: As a performance baseline for our experiments we include a model that always predicts the mean of all past observations without taking into account any additional information:

$$y_{pred} = \frac{1}{n} \sum_{i=1}^{n} y_{train,i} \; ,$$

where n is the number of charging events in the training set and y is the required energy.

(2) **Ridge Regression**: As a second model we apply ridge regression to our problem since it is a very simple method that works well if the data has a strong linear component. Compared to linear regression it is less likely to overfit the training data because of an additional weight regularization term (usually L_2 regularization). However, if the relationship between input variables and target is not linear, ridge regression is not able to find a good fit. The cost function of ridge regression cannot be customized, except for the strength of the regularization term.

(3) **XGBoost**: Furthermore, we use the XGBoost implementation by Chen and Guestrin (2016) of Boosted Gradient Trees because it has shown superior performance on structured datasets compared to other models which is proven by many data science competitions where it significantly outperformed other models (Chen and Guestrin, 2016). We specifically choose the XGBoost implementation because of its efficiency and support of user-defined objective functions.

(4) **Other Models**: Apart from the previous models we also applied other common machine learning models such as decision trees, k-nearest-neighbors, and random forests. However, preliminary experiments showed that they do not have any significant advantage on our prediction task and result in similar or worse results. Therefore, we exclude them from the rest of our report.

After processing the raw time series data as described in Section 2.2 we have a set of input features X extracted from the time series and the target feature y which is the required energy. It is very likely that the amount of energy charged at a certain event does not only depend on input features of that event but also on features of previous events as a user will likely charge more energy if the last charge happened one week ago compared to a person that already charged the car the same day. Therefore, we use a sliding window of events as input to the machine learning models to include temporal dependencies between charging events. A sliding window includes the input features of the past T events where T is the size of the window. It also includes features of the current event that are available at the time a user plugs in his car such as the date and time of the start of the charging event or the number of previous charges on the same day. Our objective here is then to approximate the relationship:

$$f(X_t, (X_{t-1}, y_{t-1}), ..., (X_{t-T}, y_{t-T})) = y_t \, .$$

3.1.2 Conditional Probability Model

As discussed at the end of Section 2.4, the most promising predictors of our prediction are categorical features extracted from timestamps, e.g. `hour`, `day` and `dayofweek` etc, as well as engineered features such as 4-hour-cluster and 8-hour-cluster, and possibly accumulative connection counts of the current day or week. Together with the assumption that the target variable, i.e. required energy, is a random variable taking values from 0 to its capacity by the law of some unknown probability measure, previous analysis allows us to consider following statistical framework, outlined in Wolf (2018).

Let $\mathcal{X}$ and $\mathcal{Y}$ be the space of input and output, P the corresponding probability measure over the product space $\mathcal{X} \times \mathcal{Y}$, and $\mathcal{A} : \cup_{n\in\mathbb{N}}(\mathcal{X} \times \mathcal{Y})^n \to \mathcal{Y}^{\mathcal{X}} : S \mapsto h_S$, a learning algorithm. S represents the training data, the input of a learning algorithm, and its corresponding output, Borel measurable function $h_S : \mathcal{X} \to \mathcal{Y}$ is a hypothesis that aims at predicting $y \in \mathcal{Y}$ from an arbitrary $x \in \mathcal{X}$. Given a loss function $L : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$, we are looking for a learning algorithm $\mathcal{A}$, precisely a set of hypothesis $\{\hat{h}_S \in \mathcal{Y}^{\mathcal{X}} : S \in \cup_{n\in\mathbb{N}}(\mathcal{X} \times \mathcal{Y})^n\}$ such that for any S, $\hat{h}_S(X)$ minimizes the expected loss, called risk, i.e.

$$\hat{h}_S := \min_{h_S} \mathbb{E}[L(Y, h_S(X))] \, ,$$

where (X, Y) a random variable of $\mathcal{X} \times \mathcal{Y}$ by the law P.

We use minimum instead infimum, because in practice, only finite number of elements in $\mathcal{Y}$ are considered, thus there exists a minimizer.

For discrete X, one can construct the minimizer point-wise, by defining

$$\hat{h}_S(x) := \min_{h_S(x)\in\mathcal{Y}} \mathbb{E}[L(Y, h_S(X))|X = x] \, .$$

Clearly this construction is well-defined.

Since the P is unknown, the risk cannot be evaluated directly, a common approach is to use empirical risk as an approximation, which leads to the following prediction,

$$y_{pred}(x) = \min_{h_S(x)\in\mathcal{Y}} \frac{1}{m} \sum_{i=1}^{m} L(y_i, h_S(x)) \, .$$

where $\{(x, y_i) : i \in [m]\}$ is the set of pairs of data points in training data that have the same input value.

Depending on the loss function, one could have different forms of predictions:

- L_2-loss, i.e. $L(y, y') := (y - y')^2$, this is equivalent to the mean model, which leads the estimator to provide the mean as prediction.
- Quantile Regression Loss (also known as Pinball loss), i.e. $L(y, y') := \rho_\tau(y - y') = (y - y')(\tau - 1_{(y-y'<0)})$ where $\tau \in [0, 1]$ is the desired percentile form given distribution function F_Y of Y (Koenker, 2005). The minimizer of its empirical expected loss is the τ-th quantile of the F_Y.

This approach gives the freedom of using customized loss functions, possibly even an ensemble of them, keeping the confidence of predictions in control, by comparing them with different quantiles. In our case, we designed a loss function that reflects the user needs better in business scenarios, which will be introduced in the subsequent section.

Beyond that, during the training, we use all available past events to predict a new event. We do not use the sliding window approach, as the conditions would become two specific and we would not have enough data points that match the given conditions for a new event. Considering cases when there is not enough data points (few, or none) from the past to meet the levels of all conditions, we drop conditions one by one from a predefined list, which is arranged by their feature importance coefficients produced by XGBoost with a sample pool, starting with a least important condition in the remaining list, until a threshold θ on the size number is passed.

Similar to the previous section we want to approximate the relationship between input features X and required energy y. For the conditional probability model we only use categorical features as predictors, because, as discussed in Section 2.4, the metric features are not good predictors for our case. Furthermore, we do not apply sliding window on the filtered past events with conditions, as we might not have enough data points that match the given conditions. And the data time span of 2 year and 5 months should be long enough to capture the charging behavior, but not too long to overlook recent significant behaviour changes.

Therefore, the objective in this approach to approximate the relationship:

$$f(X_t, (X_{t-1}, y_{t-1}), ..., (X_1, y_1)) = y_t \, .$$

3.2 Evaluation

In order to address our three project objectives we defined multiple evaluation strategies and metrics in Section 3 to test and compare our different methods.

3.2.1 Error Function

First, it is important to define how the error between prediction and target variables should be measured. Predicting the amount of energy needed for a full charge of an EV is a regression problem. We have several requirements that need to be incorporated into our error function in order to accurately reflect the real-life consequences of a wrong prediction:

- Too low predictions should be penalized more than too high predictions, as the most important objective is that a user has a fully charged car at the specified time. Thus, too low predictions would lead to an insufficiently charged battery whereas too high predictions would merely cause a suboptimal charging plan.

- The error should be in relation to the battery size of the EV, as for an EV with a small battery size a difference between prediction and target of 2kWh already has an impact whereas for a car with a battery size of 40kWh that difference will be hardly noticed.
- Greater differences between target and prediction should be penalized significantly higher than small differences.

Common evaluation metrics for regression problems such as mean squared error (MSE) or mean absolute error (MAE) are only suited up to a certain degree for our problem as they equally penalize too low and too high predictions.

Therefore, we propose a customized error function called asymmetric quadratic error (AQE) which is inspired Saxena et al. (2008), who predict the occurrence of engine failures. In their case late predictions are significantly worse than early predictions as they could lead to severe engine damages which is why they used an asymmetric exponential error function. However, an exponential error function has the significant drawback that one wrong prediction may dominate the whole error score. Therefore, we propose an adopted version that is only quadratic in the difference between prediction and target:

$$AQE = \begin{cases} (\frac{d}{a})^2 & \text{for } d < 0 \\ (\frac{d}{b})^2 & \text{for } d \geq 0 \,, \end{cases}$$

where d is the difference between predictions and targets normalized by the estimated battery size of the corresponding car:

$$d = \frac{y_{target} - y_{pred}}{\text{capacity}_{est}} \,,$$

and a and b control the asymmetry. We choose following values for a and b so that too low predictions are penalized significantly more:

$$a = 0.03, \quad b = 0.07 \,.$$

Figure 14 depicts the error as a function of the normalized difference between prediction and target.

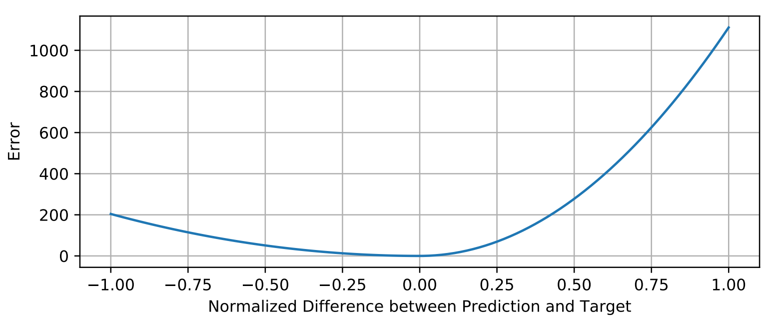


Figure 14: Asymmetric quadratic error function

The AQE fulfills all our requirements as it penalizes too low predictions more than too high predictions and also normalizes the difference by the estimated battery size of the tested car. Furthermore, differences are quadratically penalized.

3.2.2 Training Strategies

There are multiple possibilities to train a model on the given dataset depending on what data is used for training. We introduce three different training strategies using a different amount of data:

(1) **Single User**: The first method uses one separate model for every user and trains it only on past data of that user. This strategy has the benefit that the model only learns the charging patterns of one specific user and the corresponding EV. However, it has the drawback that it can only be applied once enough data for a certain user is gathered. This is a significant drawback as for a user who just signed up for the sonnenCharger, no past data is available. Furthermore, it does not incorporate data from other users with possibly similar charging patterns.

(2) **All Users**: The second method is to train one single model on the whole dataset and then do predictions for all users using this model. This approach has the advantage that the model can do predictions as soon as a new user signs up for the sonnenCharger. Furthermore, more training data is available which could improve the generalization error of the model. This approach has the drawback that the model is trained on both data from cars with small batteries and cars with bigger batteries which might imply different charging behaviors as cars with smaller batteries would be charged more frequently. In addition, different users habits also lead to different charging patterns.

(3) **Similar Users**: We propose a new training method where users are divided into clusters based on the estimated battery size of the corresponding EV. We manually set the thresholds to 10, 15, 18, 22, and 40 grouping cars with similar battery sizes together as depicted in Figure 15. This method combines the advantages of both previous approaches as it can be used as soon as a user signs up for the sonnenCharger and the model is only trained on cars with similar battery sizes.

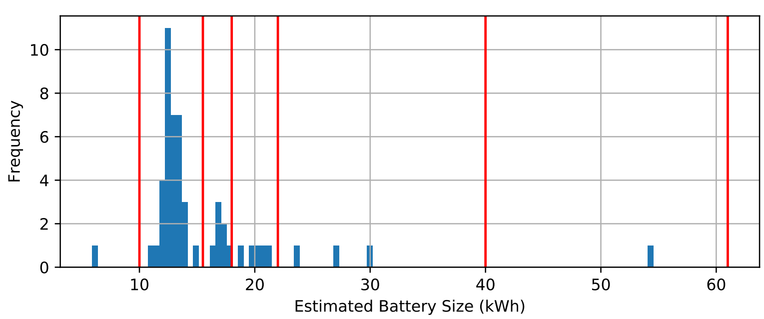


Figure 15: Cluster thresholds based on estimated battery size

3.2.3 Experiments

We designed several experiments to address our initial project objectives.

(1) **Which are the best methods to predict the required energy?**

In general, machine learning models can be evaluated using methods such as last block evaluation or k-fold cross-validation. However, a train-test set split only makes use of a limited part of the data and the time series data invalidates one of the main assumptions of cross-validation, namely that the train and test sets are sampled from the same distribution which is not the case for non-stationary time series.

Therefore, we use an approach called rolling-origin-update evaluation where the test set has a fixed size and the origin of the test set moves forward in time throughout the evaluation as described by Bergmeir and Benítez (2012) and depicted in Figure 16. This evaluation method better reflects how the algorithm will be used in production since models can be periodically retrained as more data becomes available. We evaluate all models and training strategy combinations using that approach.

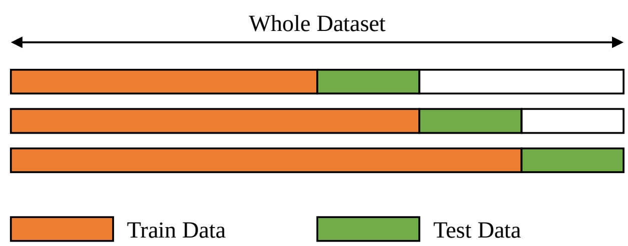


Figure 16: Rolling-origin-update evaluation

(2) **How much data is needed to accurately predict the required energy?**

It is important to evaluate how each model performs if only a certain amount of data is available. If one model performs better with less data it may be feasible to apply that model once a user signs up for the sonnenCharger and switch to a different model which performs better as more data becomes available.

Therefore, we propose a second evaluation method which uses rolling-window evaluation where the size of the training set is kept constant for all evaluations. This evaluation is then repeated for different training set sizes as depicted in Figure 17. Again we apply that evaluation method on all methods and training strategy combinations.

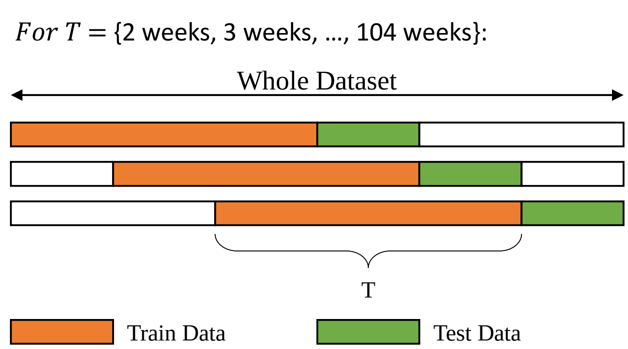


Figure 17: Rolling-window evaluation

(3) **How to deal with cases where the required energy cannot be reliably predicted?**

Lastly, we also want to take a look at how the prediction quality differs among different users to see if our approaches work for all users or whether there are users that are not easily predictable using our methods.

Therefore, we calculate the error using rolling-origin-update evaluation for each user separately and compute the correlations between the AQE and several user specific metrics to investigate for which users our methods are applicable.

3.3 Hyperparameter Tuning

An important part of improving the performance of both machine learning models and the conditional probability model is to tune the involved hyperparameters. Therefore, we ran a preliminary hyperparameter search on the most important hyperparameters of our methods using the training strategy with single users as described in Section 3.2.2. We tune the following hyperparameters:

(1) **Number of past events to include as input to machine learning models**: For the number of past events to include as input to the model we tested windows with a size $T \in \{1, ..20\}$ using XGBoost and a fixed set of input features. While using a greater window may lead to higher accuracy because more information of past charges is available it may also lead to overfitting since more features are used.

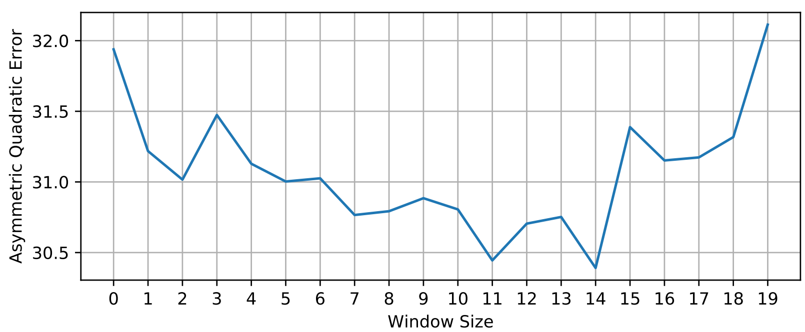


Figure 18: Impact of window size T on AQE

The results depicted in Figure 18 show that with increasing window size the error goes down until it reaches its minimum at $T = 14$. Overall, the difference in error of different window sizes is comparably small which indicates that past events do not have a significant impact on the required energy.

(2) **Features used for machine learning models and the conditional probability model**: The hyperparameter tuning for feature selection is run separately for machine learning models and the conditional probability approach as the conditional probability model only supports discrete features. We use random search to find the best feature combination as it is more efficient than grid search and often leads to similar results as shown by Bergstra and Bengio (2012). In both cases we run the search for 10000 iterations to find the best feature combination. For the conditional probability model we also test different permutations as the order of features specifies which feature is

dropped first in case there are less datapoints than the given threshold. The best found feature combinations for the respective models are:

Machine Learning models: *month, hour, day of week, day of year, mean power of stage 2 and 4, duration of stage 1 and 2, energy charged in stage 2 and 4, time difference to last charge, full charge, required energy*;

Conditional Probability model: *hour divided by eight/four, hour, number of previous charges during week, month, day of week, season, number of previous charges during day.*

(3) **Thresholds for Conditional Probability model**: Lastly, we test the threshold θ for the minimum number of datapoints needed to make predictions using the conditional probability model and a fixed set of discrete features. As training only on single users has much less datapoints available than training on similar users or all users, we run the hyperparameter search for all training strategies separately with $\theta \in \{0, .., 300\}$.

The results for all training methods are depicted in Figure 19. The AQE decreases drastically for all strategies at first and then slowly converges. At some point the error starts to increase again. The best results are $\theta = 15$ for single users, $\theta = 40$ for similar users and $\theta = 185$ for all users.

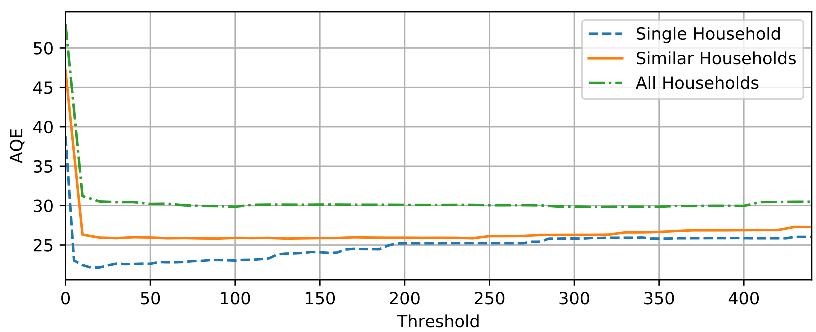


Figure 19: Impact of threshold θ on AQE using different training strategies

Based on our hyperparameter search, all future experiments are evaluated using the best combination of window size and features for machine learning models, and thresholds and features for the conditional probability model respectively.

4 Results

This section presents the results of the experiments described in Section 3. First, the general performance of each method was evaluated. Afterwards, we tested how much data is needed for each approach and lastly we tested whether there are significant differences in prediction accuracy between different users.

4.1 Which are the best methods to predict the required energy?

We evaluated every method using the rolling-origin-update evaluation on every training strategy described in Section 3.2.2. First, the models were trained on single users. Table 1 shows the results from all test runs using different models. XGBoost obtained the best results for both MAE and MSE and the second best AQE. The conditional probability model achieved the best AQE although it has the worst MSE and MAE errors which is most likely due to its optimization for AQE. Ridge regression achieved worse error scores than the baseline mean model.

Table 1: Results of all methods trained on single users

Method	MSE	MAE	AQE
Mean Model	9.97	2.55	39.79
Ridge Regression	12.99	2.65	51.45
XGBoost	**9.46**	**2.43**	26.57
Conditional Probability Model	14.24	2.98	**23.99**

Afterwards, the models were trained on all users. The results are listed in Table 2. XGBoost achieved the best AQE, whereas ridge regression obtained the best results for MAE and MSE. This is most likely due to ridge regression using MSE as objective function while XGBoost is trained on AQE. The conditional probability model obtains significantly worse results compared to training it only on a single user. Although, MSE and MAE of the conditional probability model are worse than the mean model it still achieves the second best AQE.

Table 2: Results of applied methods trained on all users

Method	MSE	MAE	AQE
Mean Model	13.11	3.08	45.17
Ridge Regression	**9.60**	**2.47**	40.87
XGBoost	12.94	3.03	**23.69**
Conditional Probability Model	20.04	3.71	33.33

Lastly, the models were trained on data of similar users. The results are shown in Table 3. Most of the results are slightly better than training the methods on all users. Again, ridge regression obtained the best results using MAE and MSE whereas XGBoost has the best AQE. Compared to all other models, the conditional probability model obtained significantly better results if it is trained on similar users in comparison to training it on all users.

Table 3: Results of applied methods trained on similar users

Method	MSE	MAE	AQE
Mean Model	12.88	3.05	45.47
Ridge Regression	**9.56**	**2.47**	41.27
XGBoost	12.34	2.95	**23.35**
Conditional Probability Model	18.17	3.51	28.23

Ultimately, the conditional probability model performs best on our error measure if it is trained on single users. The second best results are obtained by XGBoost trained on similar users. Ridge regression and the baseline model perform significantly worse on any training strategy using the AQE as evaluation metric. Also, it becomes apparent that good MAE and MSE results do not necessarily indicate that the model also performs well on our customized error metric.

4.2 How much data is needed to accurately predict the required energy?

After evaluating the general performance of all methods, we evaluated how every model performs if only a certain amount of data is available. We evaluated it with models that are trained on data of the same user, data of similar users, and data of all users using the rolling-window evaluation.

The results of models trained on a single user are depicted in Figure 20. The conditional probability model performs best until 100 weeks of training data are used where it is outperformed by XGBoost. XGBoost performs worse than the mean model if only a few weeks of training data are available. However, as more data becomes available its performance gets closer to the results of the conditional probability model. It is also interesting to note that the performance of the conditonal probability model starts to decrease after around 60 weeks of available training data. This may be due to changing user habits after a certain amount of time. Ridge regression performs significantly worse than all other methods.

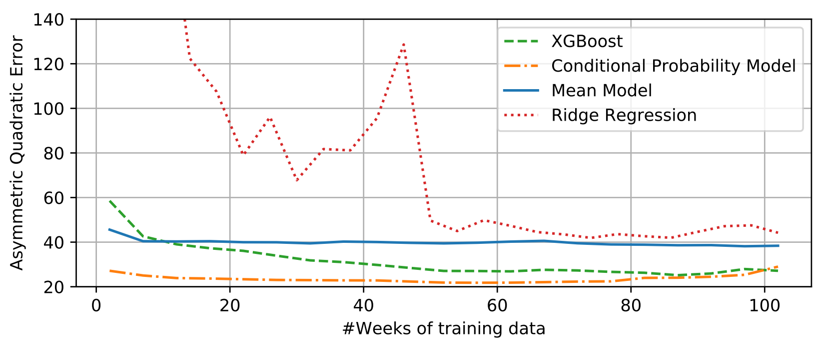


Figure 20: AQE of predictions of models trained on a single user

Figure 21 depicts the results for models trained on all available users. XGBoost outperforms the conditional probability model with a great margin in error. Furthermore, the performance of XGBoost takes less time to converge compared to the case where it is only trained on single users. Also, compared to training on single users, the conditional probability model generally obtains worse scores whereas XGBoost obtains better scores. Furthermore, the results clearly show that at some point the performance becomes worse for XGBoost and the conditional probability model if more data is used for training which indicates that the time series is non-stationary and that a users charging patterns change.

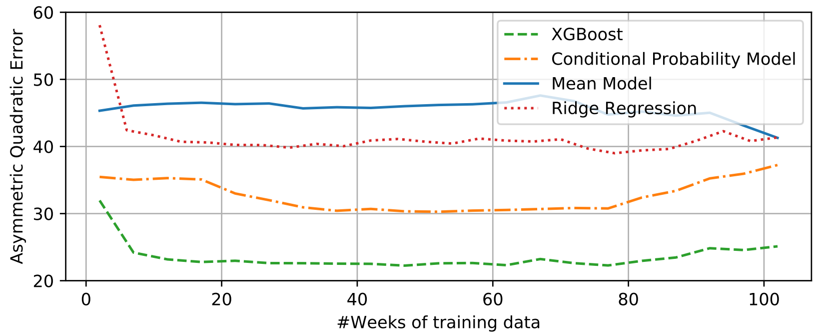


Figure 21: AQE of predictions of models trained on all users

Lastly, Figure 22 depicts the results of models trained on similar users. Except for the case where only a few weeks of training data are available, XGBoost outperforms the conditional probability model significantly. As already visible in previous results, the performance of both XGBoost and the conditional probability model gets worse at some point. Furthermore, the AQE is slightly better than the model trained on all clusters for XGBoost but convergence is slightly slower. As for training on all users, ridge regression performs slightly better than the mean model if enough data is available but still significantly worse than the other tested methods.

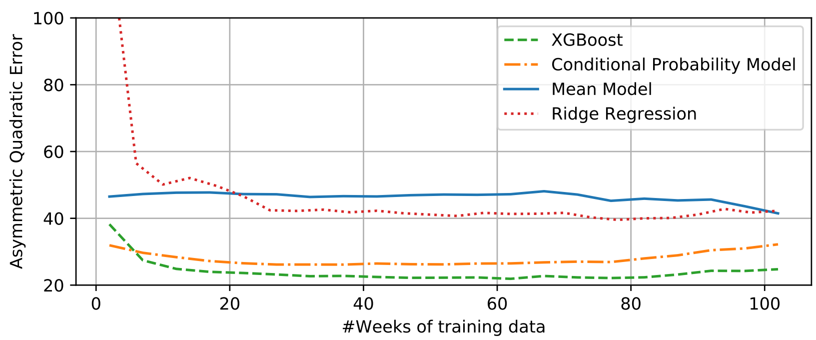


Figure 22: AQE of predictions of models trained on similar users

In conclusion, the conditional probability model performs best if trained only on single users irrespective of the amount of available data. Furthermore, XGBoost obtains significantly better results if trained on similar users which are however slightly worse then the results obtained by the conditional probability model trained on single users.

4.3 How to deal with cases where the required energy cannot be reliably predicted?

As a last step, we evaluated how prediction accuracy differs between different users. Figure 23 depicts the distribution of AQE errors of different users using XGBoost trained only on past data of the respective user.

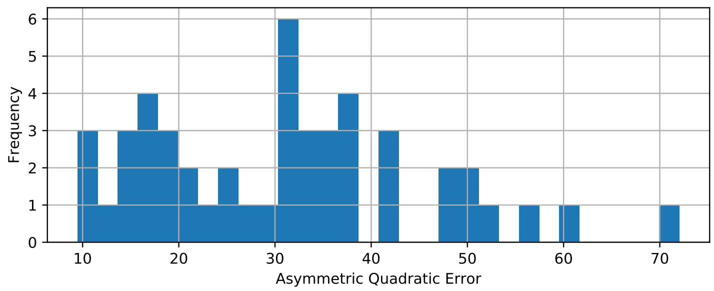


Figure 23: Distribution of AQE of different users

The error scores range between 10 and 70 which indicates that the prediction quality varies a lot across different user and which is why we further investigated what influences the accuracy of our predictions and which users are easier and harder to make predictions for. Figure 24 shows the targets and predictions using XGBoost of a user with high AQE. The red line indicates accurate predictions. The model does a poor job in predicting the required energy and mostly predicts values between 3 and 5 irrespective of the actual target value which indicates that the model can not infer how much energy is required for the next charge.

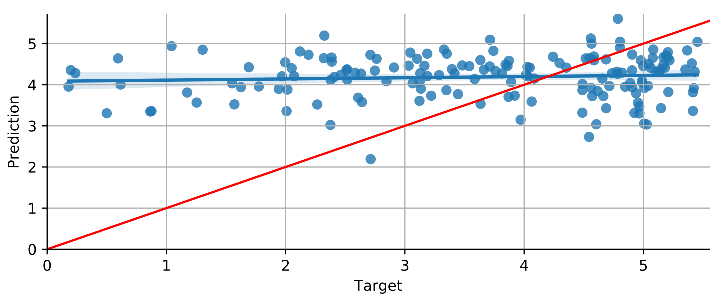


Figure 24: Predictions and targets of required energy of user 26

Figure 25 shows the same plot for a user with low AQE. Most points lie either on or near the red line which indicates perfect predictions. Also, the linear regression model fitted to the data points (indicated by the blue line) is almost the same as the red line. It is also important to note that most predictions are slightly above the red line as this leads to a better error because of the asymmetry in the error function. The main difference between these two users is that user 26 has an estimated battery size of 6.2kWh and user 114 of 13.1kWh.

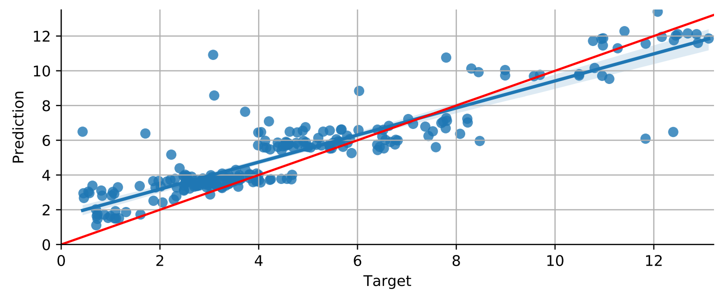


Figure 25: Predictions and targets of required energy of user 114

To further investigate which factors have an impact on the prediction accuracy we calculated several user specific features and their correlations with the AQE. The correlations in Table 4 indicate that the predictions are best if a user has a car with greater battery, but does charge less energy at each charge. Furthermore, charging more frequently has a negative impact on the overall error and it is of high importance that enough charging events for that user are available for training of the model.

Table 4: Pearson correlation coefficients between AQE and user specific features

Feature	Correlation with AQE
Charging frequency	0.132
Battery capacity	-0.349
Average amount of energy charged at each charge	0.191
Number of valid charging events	-0.359

The negative correlation between battery capacity and AQE might be due to the normalization in AQE. The larger the battery size is, the more time the EV would need to charge to full. Since full charges occur most likely during night, and the span of night hours is fixed, when only normalized required energy is evaluated, there would be less range variation for larger battery size than for small battery size. The negative correlation between average charged energy and AQE is probably due to that the more a user charges at home, the more information we have about this user's charging behavior, thus less prediction error occurs.

Although, the correlation coefficients indicate for which users our predictions work well, the coefficients are still only moderate. Therefore, we have to conclude that without further research they can only serve as an indication whether our predictions work well and not definite proof.

4.4 Resulting Approach

Based on these results we designed an algorithm for the prediction of the required energy once a new user signs up for the sonnenCharger. First, we always use the estimated battery size of that user as our prediction as it is the least risky prediction and always leads to a full charge. As our prediction accuracy differs significantly among different users, we first test our predictions before actually applying them. Thus, we evaluate the predictions of

XGBoost trained on similar users on that user for a certain amount of time since XGBoost obtained the best results with that training strategy and it is directly applicable. If the AQE is low we apply the predictions of XGBoost on that user instead of using the battery size. If the AQE is high we continue to use the battery size. After a certain amount of time as more data for that user becomes available we evaluate the conditional probability model trained on that single user as it has a better AQE than XGBoost trained on similar users, if enough data is available. If the AQE is lower than using XGBoost we switch to the conditional probability model. If it is higher we continue to use XGBoost or the battery size.

We extend this approach by introducing a confidence parameter α that controls how much our predictions influence the final prediction for the required energy:

$$y_{final} = \alpha \cdot y_{pred} + (1 - \alpha) \cdot \text{capacity}_{est}$$

where y_{final} is our final prediction, y_{pred} is the prediction using our models and capacity_{est} is the estimated battery size. Low α implies that we mainly rely on our estimate of the battery size whereas high α implies that we mainly use our predictions. There are several ways to set α:

1. **Based on AQE**: One way to set α is to choose it anti-proportional to the resulting AQE for a single user since a worse error indicates inaccurate predictions so that it is safer to rely on the battery size estimate instead.

2. **Based on user preferences**: Another way is to either let a user specify his confidence level, or develop algorithm to determine the confidence level for user based on standardized preference survey. If a user prefer absolute confidence, α is set to 0. If he is fine with lower confidence α can be set higher.

This approach allows us to incorporate our predictions depending on their quality or a users preferences which overall leads to more reliable predictions.

5 Conclusion & Outlook

The goal of our project at sonnen was to predict the required energy once a user plugs in his car to the sonnenCharger in order to optimize the charging plan for the use of solar energy. To reach this goal we investigated three project objectives: which models work well on our problem, how much data is needed, and how the performance differs among different users. As a result we found out that the conditional probability model performs best if trained on a single user and XGBoost obtains the best results if trained on similar users. Furthermore, the performance of our models is different depending on the user and we could find indicators for which users our models work well but no definite proof.

Since the sonnenCharger only started shipping in April 2018, we did not have enough data to apply our approach on data from sonnen. However, we implemented an adapter that transforms the data from sonnen to the same format we used for Dataport. Therefore, one can just switch the adapter for Dataport with our adapter for sonnen and evaluate our algorithms as soon as enough data becomes available. As the Dataport dataset contains recent data from normal users, we expect that our models achieve comparable results on the sonnen data.

Additionally, during development, we identified future research directions and a few points where the current model might be improved using additional data only available for the sonnen dataset:

- **Clustering:** As described in Section 3.2.2, we divided the batteries into clusters based on estimated battery capacity. This is optimal for our investigation given the limited nature of the dataset used, but for sonnen's implementation other clustering variables could prove more effective, such as actual battery capacity, EV model, or user demographics data.

- **Feature extraction:** As mentioned in Section 1.2.3, the distinction between prematurely ended and complete charges can be made in a more simple and robust way through the use of the charger pilot signal. Actual battery capacity will replace the estimated capacity. Also, EV model would give precise information about its steady state power. Furthermore, features of first-hand collected user preferences could be included as predictors.

- **Customized error function:** Similar to the confidence parameter α described in Section 4.4 that can be set using confidence collected from user preferences, one can also include this preference in the training of our models by updating the steepness parameters of the AQE separately for each user and the corresponding preference.

References

C. Bergmeir and J. M. Benítez. On the use of cross-validation for time series predictor evaluation. *Information Sciences*, 191:192–213, 2012.

J. Bergstra and Y. Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(Feb):281–305, 2012.

G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pages 785–794. ACM, 2016.

I. E. Commission et al. Iec 62196-2: 2016. *Plugs, Socket-Outlets, Vehicle Connectors and Vehicle Inlets–Conductive Charging of Electric Vehicles–Part*, 2, 2016.

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*, 81:36 – 56, 2017.

S. Dhameja. *Electric vehicle battery systems*. Elsevier, 2001.

T. Franke and J. F. Krems. Understanding charging behaviour of electric vehicle users. *Transportation Research Part F: Traffic Psychology and Behaviour*, 21:75–89, 2013.

P. Kasten, J. Bracker, M. Haller, and J. Purwanto. Electric mobility in europe–future impact on the emissions and the energy systems, 2016.

R. Koenker. *Quantile regression*. Econometric Society Monographs. Cambridge University Press, 2005.

K. Morrow, D. Karner, and J. Francfort. Plug-in hybrid electric vehicle charging infrastructure review. *US Department of Energy-Vehicle Technologies Program*, 34, 2008.

J. Quirós-Tortós, L. F. Ochoa, and B. Lees. A statistical analysis of ev charging behavior in the uk. *2015 IEEE PES Innovative Smart Grid Technologies Latin America (ISGT LATAM)*, pages 445–449, 2015.

A. Robinson, P. Blythe, M. Bell, Y. Hübner, and G. Hill. Analysis of electric vehicle driver recharging demand profiles and subsequent impacts on the carbon content of electric vehicle trips. *Energy Policy*, 61:337 – 348, 2013.

A. Saxena, K. Goebel, D. Simon, and N. Eklund. Damage propagation modeling for aircraft engine run-to-failure simulation. In *Prognostics and Health Management, 2008. PHM 2008. International Conference on*, pages 1–9. IEEE, 2008.

T. Stuart, F. Fang, X. Wang, C. Ashtiani, and A. Pesaran. A modular battery management system for hevs. Technical report, SAE Technical Paper, 2002.

C. Toepfer. SAE electric vehicle conductive charge coupler, SAE J1772. *Society of Automotive Engineers*, 2009.

J. Wang, C. Liu, D. Ton, Y. Zhou, J. Kim, and A. Vyas. Impact of plug-in hybrid electric vehicles on power systems with demand response and wind power. *Energy Policy*, 39(7): 4016 – 4021, 2011. Special Section: Renewable energy policy and development.

C. Weiller. Plug-in hybrid electric vehicle impacts on hourly electricity demand in the united states. *Energy Policy*, 39(6):3766 – 3778, 2011.

M. M. Wolf. Lecture notes on mathematical foundations of machine learning (growing), June 2018.

E. Xydas, C. Marmaras, L. M. Cipcigan, N. Jenkins, S. Carroll, and M. Barker. A data-driven approach for characterising the charging demand of electric vehicles: A uk case study. *Applied Energy*, 162:763 – 771, 2016.