

# Electric Vehicles in Energy Communities: Investigating the Distribution Grid Hosting Capacity

Albert Ludwig University of Freiburg

Daniil Aktanka

August 22, 2022







## 2 Literature Review

## 3 Goal and Method

### 3.1 Goal

Gotta get themn simulations on

### 3.2 Data processing

Data processing was done entirely in python with a heavy use of the pandas module. For the sake of clarity, most of the code was split between a python class file and jupyter notebooks. The python class file consists of two classes, one for handling data processing and another for operations based on the pandapower module. The following information describes the functionality of the python class file for data processing. This section covers crucial data processing steps and functions tailored to the household dataset.

**Data import and segmenting:** The raw dataset is imported as dataframe without additional options. Dropping the rest, we keep only three columns: `DE_KN_residential1_grid_import`, `DE_KN_residential2_grid_import` and `utc_timestamp`. Since we will be performing conditional time-based operations, we set the timestamp column as index for ease of use. While it is possible to front-load a lot of data processing functions at this step—such as parsing datetimes—it is not recommended due to unnecessary computation time. A better approach is to segment the data for piecewise processing and function testing, whereby it would be possible to iterate computations over the entire data set in the future. Therefore, we split the imported dataframe (of a little over a million data points) into a list of smaller dataframes (each 10000 datapoints long, with residual last dataframe being a bit smaller).

**Datetime parsing function:** This function converts a segment of the imported data into a specific, time-indexed dataframe. First, we parse datetime index, specifying the format as "Year-Month-Day Hour:Minute:Second", after which we convert from UTC to Berlin time—the local time of the recorded dataset. While it is possible to import data with local time column `cet_cest_timestamp` without the need of conversion, it is not recommended for pandas 1.4.3 since that will cause errors in datetime operations based on my experience. Second, we take the difference between consecutive rows. This is done in order to obtain minute-wise energy changes, since the household dataset only tracks commulative energy values. If we use the `pandas.diff()` function, we must also drop the resulting first row, since it's a NaN row.



## 4 Results and Discussion

### 4.1 Results

#### 4.1.1 Test network

#### 4.1.2 European network

### 4.2 Suggested improvements





