# Residential Electric Vehicle Charging Characterization via Behavior Identification, Vehicle Classification, and Load Forecasting

CS 229 Project Final Report | Category: Physical Sciences

Antonio Aguilar (antonioa), Justin Luke (jthluke), Robert Spragg (spragg)

***Abstract*— This report presents three main findings in the characterization of residential electric vehicle (EV) charging: unsupervised charging behavior identification, classification of car models, and forecasting of charging load. Characterization is essential for electric utilities, policymakers, and charging equipment companies to plan infrastructure investment and inform demand-response programs to better utilize renewable energy. From a high-resolution dataset of 75 homes with EVs, charging sessions are extracted and features such as power rate, session energy, charging tail, and time variables are derived. Firstly, k-means and DBSCAN clustering are used to identify user behavior, presenting a potential automated method to distinguish between full charge sessions and user-terminated sessions, as well as between programmable and non-programmable charging sessions. Secondly, the derived features of battery capacity and maximum charge rate are used as inputs to a semi-supervised Gaussian Mixture Model to classify vehicle models. Thirdly, various regression methods are tested to forecast aggregate day-ahead and average week-ahead hourly EV load. Compared to a naïve persistence model baseline, a neural network was found to have best overall performance on the test split data averaged across both experiments.**

## I. Introduction

Electric Vehicles (EV) can serve as mobile energy storage assets for the electricity grid by performing demand response, participating in ancillary services, and supporting renewable energy integration. However, EVs also have the potential to adversely impact the grid by further amplifying the *solar over-generation duck curve* and by reducing the lifespan of existing power transformers [1] [2]. The optimization of charging schedules within users' space of possibilities will be a determining factor in the size of the environmental and energetic footprint of EVs.

We deem it critical to understand the charging behavior in a residential environment, as it has been shown that the majority of charging occurs at home [3]. While the flexibility for EVs to provide grid services has been studied in the public and commercial setting, representation of EV charging in residential environments is less understood. The state of California, for instance, plans to have over 5 million electric vehicles in use by 2030. This makes residential smart charging ever more critical to ensure EVs are supporting the grid and helping achieve the state's goal of reaching 100% renewable energy by 2045.

We perform our characterization on high-resolution (1-minute) residential electric vehicle charging data obtained from 2016-2017 provided by Pecan Street, a research test-bed in the neighborhood of Mueller in Austin, Texas. This data was collected from 1,115 active homes and businesses, which includes 75 electric vehicle owners. From the raw time-series EV charging data, we extract over 30000 charging sessions and derive various features to input to our machine learning models, which produce three main findings presented in this report.

Firstly, we characterize the different charging patterns that exist in the data, using unsupervised clustering algorithms. We use k-means clustering with heuristically estimated battery capacity and maximum charge rate of each home as inputs to output clusters which may distinguish full charging sessions from prematurely user-terminated sessions. We also use density-based spatial clustering of applications with noise (DBSCAN) with hour of day for the start and end times of sessions as inputs to output clusters which potentially identify users who have programmable charging equipment.

Secondly, we perform semi-supervised classification of vehicle models using a Gaussian mixture model (GMM) in the space of discovered characteristics. We are able to leverage the labeling of a subset of electric vehicles in the data set in order to infer the make and models of other unlabeled cars. Heuristically estimated battery capacity and maximum power rate for each home were used as model inputs to output a vehicle model label. Vehicle model classification is useful for customer segmentation and identification of eligible smart charging algorithms, since charging control can vary by manufacturer and model.

Thirdly, we tested and tuned various regression models to forecast hourly-resampled aggregate residential EV charging demand. In our first experiment, past seven days of demand, one-hot encoded day of week, and one-hot encoded month of year were used as inputs to the models to output the coming day's demand. In our second experiment, past four 24-hour demand curves averaged over seven day periods, one-hot encoded week of year, and one-hot encoded month of year were used as inputs to the models to output the 24-hour demand averaged over the coming seven days. Forecasting of charging demand is essential to perform predictive smart charging and provide demand-response services such as load shifting, peak shaving or valley filling, ramp mitigation, frequency regulation, and renewable energy prioritized charging.

## II. Related Work

Existing literature in the field has largely focused on characterizing EV energy demand on public infrastructure and proposing optimization methods to increase the percentage of that demand that can be met with renewable sources.

Sadeghianpourhamami, et. al. analyze data from public charging infrastructure throughout the Netherlands collected between 2011 and 2015[4]. Their objective is to quantify EV flexibility characteristics in order to facilitate their integration onto the grid. They employ a DBSCAN in order to cluster users into three categories: "charge near home" (27.84%), "charge near work" (9.3%), and "park to charge" (62.86%). After analyzing seasonal effects on arrival times, observed data is compared to the results of two optimizations designed to flatten and balance EV loads with respect to renewable energy supply.

Schuller, et. al. also employ linear optimization techniques to more intelligently schedule charging sessions and therefore maximize the percentage of energy demand from EVs that can be met with renewables. Their data is not from EV charging sessions but instead from driving patters, from which they simulate charging behaviors.

Very little, if any, literature focuses solely on identification of vehicle model from charging session data. This is likely due to the lack of knowledge of EV model in available data sets from public charging stations, and the rarity of residential EV charging data that is recorded separately from the load profile of the home.

## III. Dataset & Features

Electric vehicle charging data from 75 homes in Austin, TX was collected from the Pecan Street Dataport [5]. We received observations at 1-minute intervals from the EV-specific sub-meter of the house. Example data for a single home is shown in Figure 1. Houses with incomplete or erroneous data (i.e. very little to no charges) were removed. Over 30000 charging sessions were identified across all vehicles from 2016 and 2017, after removing sessions associated with noise (peak power less than 1 kW) in the data or *topping off* of a nearly-full and plugged-in car battery (session energy less than 0.1 kWh or shorter than 2 minutes in duration).

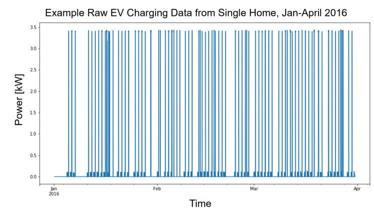


*Fig. 1: Example raw EV charging data from a single home*

A second file contained the EV model associated with each house. Some errors were identified in this data. Vehicle models that only occurred once were removed. The remaining vehicles were —Chevy Volt, Nissan Leaf (both the 3.3kW and 6.6kW versions), Ford Fusion, and Tesla Model S.

Charging sessions were extracted from the time series using a state heuristic. For each home, a histogram of charging power was created using Doane's Formula for determining binning [6]. The three most frequent power bins were identified and sorted. The largest of these three bins was determined to be associated with the ON state, whereas the lower two were associated with the OFF state and noise. Additional derived features associated with each session include: session energy [kWh], peak power rate [kW], time variables (e.g. start and end-time, day-of-week, month, week of year), and charging session tails. The charging session tail is the decrease in power rate that typically occurs as a battery approaches its fully-charged state. The tail was also extracted using a heuristic utilizing numeric differentiation of a moving-average smoothed (window size of 5 samples) power signal, determined as having a slope less than -0.1 kW/min but occurring at least in the final 90% of the session duration. An example extracted session with identified tail is shown in Figure 2.

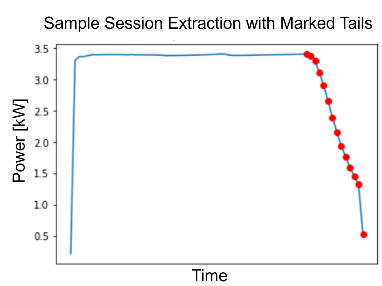


*Fig. 2: An extracted charging session, with red dots indicating the charging session 'tail'*

## IV. Methods

Our study utilized three separate methods to characterize residential EV charging:

- k-means clustering to identify full-charge and prematurely user-terminated sessions; DBSCAN to identify programmable chargers

- Semi-supervised GMM using Expectation-maximization (EM) algorithm to classify unlabelled vehicle models
- Various regression methods to forecast future EV charging load

### A. Unsupervised Clustering

k-means clustering was performed on the tails of the charging session. The k-means algorithm can be best described as finding the partition of all observations into $k \leq n$ sets $S = \{S_1, ..., S_k\}$ such that:

$$\arg\min_{S} \sum_{i=1}^{k} |S_i| \mathrm{Var}(S_i)$$

When implemented as an algorithm, randomized cluster mean initialization, L2-norm distance metric, and convergence tolerance of 0.0001 were used. For all cluster labels $c^{(i)} \in \mathbb{R}, \forall i = 1, ..., m$, data samples $x^{(i)} \in \mathbb{R}^6, \forall i = 1, ..., m$, and cluster centroids $\mu_j, \in \mathbb{R}^6, \forall j = 1, ..., k$:

1) Randomly initialize cluster centroids $\mu_j \forall j = 1, ...k$

2) Update and repeat until convergence:

$$c^{(i)} \longleftarrow \arg\min_{j} ||x^{(i)} - \mu_j||^2, \forall i = 1, ..., m$$

$$\mu_j \longleftarrow \frac{\sum_{i=1}^{m} 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)} = j\}}, \forall j = 1, ..., m$$

In order for the algorithm to cluster over tails with varying power levels and duration, we found it necessary to normalize the tails. Both the tail power and tail duration were then discretized into 0, 20, 40, 60, 80, and 100 percentiles, forming two six-dimensional features. To determine the number of clusters $k$, we utilized the "elbow-joint" method by plotting inertia (sum-squared distance of points to their cluster centroid) as a function of $k$ and choosing the $k$ where the decrease in inertia diminishes. From this $k = 3$ was determined.

DBSCAN algorithm was used to identify charging sessions associated with a programmable charger. The method is simply defined by two hyper-parameters: the minimum number of samples to constitute a cluster and the maximum distance allowable between samples of the same cluster. Any samples not meeting this criteria is marked as "noise", not belonging to a cluster. An advantage of DBSCAN is its ability to find arbitrarily-shaped clusters, robustness to noise, and no requirement to specify the number of clusters. As later shown in the results section, DBSCAN is well-suited to find any number of elongated shape cluster of sessions in the start-end time space. A disadvantage of DBSCAN can be determining its two hyper-parameters, however, with our domain knowledge of our application, we are interested in when programmable sessions constitute at least 10% of the home's total sessions and are within a 2 hour L2-distance from one another.

### B. Semi-Supervised GMM using EM

Utilities, including Palo Alto's municipal utility, do not have reliable data on which homes have an EV, or the type of EV a home has. Knowing the EV model is useful when crafting electricity rate structures, demand response programs, and assessing distribution system transformer upgrade needs.

Semi-supervised GMM EM was performed on two features—charging rate and battery capacity—in order to identify clusters which represent the same vehicle models. The EM algorithm alternates between computing the expectation of the log-likelihood evaluated using the current estimate for the parameters and maximizing that expected log-likelihood until convergence. The parameters of interest for GMM, which finds Gaussian clusters, are cluster centroids $\mu_j \in \mathbb{R}^2, \forall j = 1, ..., k$, cluster covariance matrices $\Sigma_j \in \mathbb{R}^{2\times 2}, \forall j = 1, ..., k$, and a multimodal distribution of the probability of each cluster $\phi_j \in \mathbb{R}, \forall j = 1, ..., k$. This is given unlabeled samples $x^{(i)} \in \mathbb{R}^2, \forall i = 1, ..., m$, labeled samples $\tilde{x}^{(i)} \in \mathbb{R}^2, \forall i = 1, ..., \tilde{m}$, latent labels $\tilde{z} = j, \forall j = 1, ..., k$, and weight $\alpha \in \mathbb{R}$ for importance of semi-supervised samples:

Semi-supervised E-step:

$$w_j^{(i)} = p\left(z^{(i)} = j \middle| x^{(i)}; \theta\right) = \frac{|\Sigma_j|^{-0.5} \exp\left(-\frac{1}{2}\left(x^{(i)} - \mu_j\right)^T \Sigma_j^{-1} \left(x^{(i)} - \mu_j\right)\right)\phi_j}{\sum_{l=1}^{k} |\Sigma_l|^{-0.5} \exp\left(-\frac{1}{2}\left(x^{(i)} - \mu_l\right)^T \Sigma_l^{-1} \left(x^{(i)} - \mu_l\right)\right)\phi_l}$$

Semi-supervised M-step:

$$\mu_j \leftarrow \frac{\sum_{i=1}^{m} w_j^{(i)} x^{(i)} + \sum_{i=1}^{\tilde{m}} \alpha \tilde{x}^{(i)} 1\{\tilde{z}^{(i)} = j\}}{\sum_{i=1}^{m} w_j^{(i)} + \sum_{i=1}^{\tilde{m}} \alpha 1\{\tilde{z}^{(i)} = j\}}, \phi_j = \frac{\left(\sum_{i=1}^{\tilde{m}} \alpha 1\{\tilde{z}^{(i)} = j\} + \sum_{i=1}^{m} w_j^{(i)}\right)}{\alpha \tilde{m} + m}$$

$$\Sigma_j \leftarrow \frac{\sum_{i=1}^{m} w_j^{(i)} \left(\left(\tilde{x}^{(i)} - \mu_j\right)\left(\tilde{x}^{(i)} - \mu_j\right)^T\right) + \alpha \sum_{i=1}^{\tilde{m}} \left(\left(\tilde{x}^{(i)} - \mu_j\right)\left(\tilde{x}^{(i)} - \mu_j\right)^T\right) 1\{\tilde{z}^{(i)} = j\}}{\sum_{i=1}^{m} w_j^{(i)} + \sum_{i=1}^{\tilde{m}} \alpha 1\{\tilde{z}^{(i)} = j\}}$$

While most households in our data had labels for vehicle model, we chose to perform an analysis robust to even more missing labels which can be more easily replicable in a similar real-world data sets. Each time the semi-supervised EM was run, two vehicles of each model were selected from the set and used as the supervised examples. This allowed for an analysis of the robustness of the algorithm to scenarios where a home's EV model is not known.

### C. Load Forecasting Methods

Multiple forecasting methods for EV charging load aggregated across all homes were explored on two different experiments. The first involved predicting total

EV charging load for a 24-hour period given the hourly load from the preceding week, one-hot encoded day of week, and one-hot encoded month of year. The second involved forecasting the average 24-hour period hourly load expected for the following week given the averages for the 4 preceding weeks, one-hot encoded week of year, and one-hot encoded month of year. The time features help account for seasonality in the data.

We established a baseline prediction with the persistence model, which simply uses the current timestep as the prediction for the next timestep. We then implemented linear regression, support-vector machine (SVM) regression, elastic net regularization, kernelized ridge regression, and feed-forward fully-connected neural networks[7]. Using an error metric of root-mean-squared-error (RMSE), hyper-parameters for all models were tuned via a 5-fold cross validation grid search on the training split data, which is all data from January 2016 to May 2017. Their performances are then measured by their RMSE on the test split, which is all data from May 2017 - end of December 2017. We found the neural network had best average performance between the two experiments.The optimal neural network structure is shown in Figure 3.

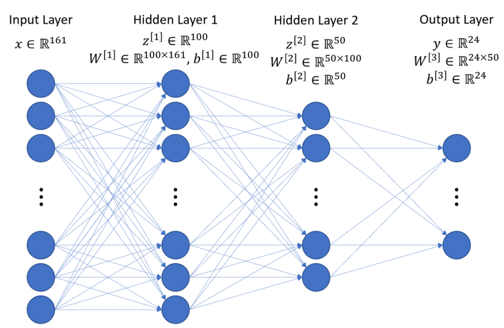


*Fig. 3: Neural network with ReLU activations was used for forecasting*

## V. Results & Discussion

### A. Behavior Identification

k-means clustering was performed for the entire set of EVs as well as separately for each model, to ensure that car-specific charging session characteristics were accounted for. Figure 4 shows the results of the clustering for a single EV. Figure 5 shows three sample sessions for each of the three typical tail profiles shown in Figure 4. Note that sessions in Cluster #0 have a gradual decrease in power over time, whereas those in Clusters #1 and #2 tend to have a sharp drop in power over a short period of time, possibly indicating a prematurely terminated session by the user. A high fraction of complete sessions vs. premature sessions could help utilities identify neighborhoods with greater charging flexibility.

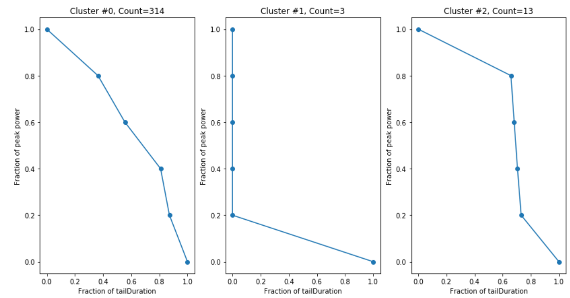


*Fig. 4: Normalized charging session k-means clustering results*

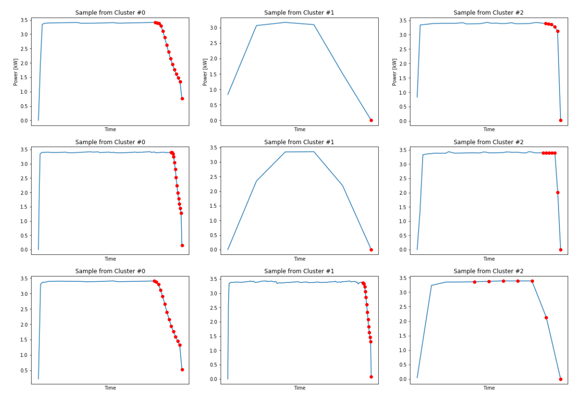


*Fig. 5: Sample charging sessions associated with each k-means cluster. The tail is identified by the red dots.*

DBSCAN was successful in identifying clusters of charging sessions for certain car owners in the hour-of-day start and end time space. We believe these to be result of programmable charging. One such result is shown below in Figure 6. This user may have programmed for their car to be 100% charged by 6AM, as seen by the cluster of sessions ending at 6AM and starting at a straight line between 2AM and 6AM depending on how much charge is required. The ”noise” labels are marked in gray, having not correspond to a dense enough cluster (as specified in Methods section).

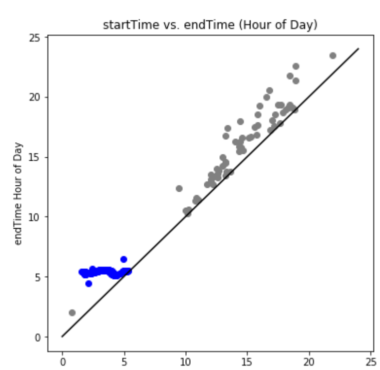


*Fig. 6: Result of DBSCAN on a single home*

| True \ Pred. | Volt | Leaf (3.3kW) | Model S | Fusion | Leaf (6.6kW) |
|---|---|---|---|---|---|
| Volt | 38 | 1 | 4 | 0 | 1 |
| Leaf (3.3kW) | 1 | 0 | 3 | 0 | 0 |
| Model S | 0 | 0 | 16 | 0 | 0 |
| Fusion | 1 | 0 | 1 | 0 | 0 |
| Leaf (6.6kW) | 0 | 0 | 4 | 0 | 3 |

*TABLE I: Confusion matrix for vehicle model classification. Precision: 0.95, 0.00, 0.57, 0.00, 0.75; Recall: 0.86, 0.00, 1.00, 0.00, 0.43*

### *B. Vehicle Classification*

The semi-supervised EM model was able to predict the correct vehicle model with approximately 80% accuracy. The accuracy varied slightly each iteration, due to the random selection of the labeled data from the set of all houses. Figure 7 shows the results of the algorithm and Table I shows the confusion matrix. The algorithm was most successful for classifying cars with many samples, but fared poorer for cars with scarce data.

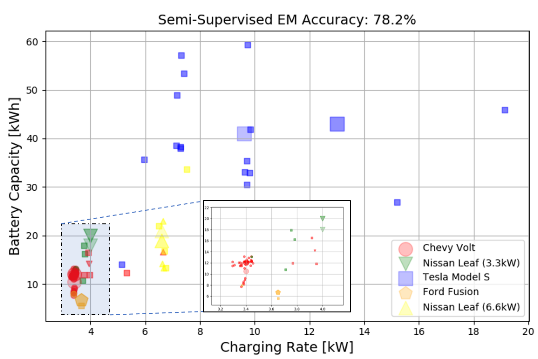


*Fig. 7: Semi-Supervised GMM EM for car model classification. Marker color corresponds to true label, shape corresponds to predicted label, and larger size corresponds to supervised labeled samples.*

### *C. Load Forecasting*

We initially undertook the task of predicting future energy demand from individual cars. However, we were unable to design a loss function that would keep our models from simply learning the null prediction given the sparsity and irregularity of charging sessions.

To remedy this, we then aggregated loads of all cars to instead predict load for the entire neighborhood. Our challenge then became to find models that would not overfit the data. This is shown by the extreme discrepancy between train and test error shown by the unregularized linear regression, as seen in the final results Table II. All other models were tuned, with regularization coefficients, learning rates, and kernel type determined by 5-fold cross validation grid search.

In the task of predicting weekly averages, the baseline predictor actually performed very well on the test split, highlighting the regularity of the data once averaged and emphasizing precaution needed to prevent overfitting. Kernelized ridge regression obtained the lowest test loss with a linear kernel and L2 penalty coefficient of $2.005 \times 10^7$. In the day-ahead prediction task, elastic net had the best balance between under and overfitting, with a regularization term of 65700 and L1 to L2 penalty ratio of 0.25. However, neural networks (architecture shown in Figure 3) had the best overall performance in both experiments. Its performance on the test split data is shown in Figure 8.

| **Model** | **Train RMSE (Daily, Weekly) m = (476, 65)** | **Test RMSE (Daily, Weekly) m = (244, 34)** |
|---|---|---|
| Persistence | (586.02, 227.96) | (533.12, 210.42) |
| Linear Reg. | (304.31, 7.15e-13) | (496.44, 437.43) |
| Elastic Net | (400.66, 164.33) | (400.13, 241.46) |
| SVM Reg. | (38.29, 109.19) | (437.05, 235.67) |
| KR Reg. | (316.51, 163.86) | (459.70, 189.65) |
| NN | (284.42, 166.42) | (419.02, 209.53) |

*TABLE II: Performance for all methods in both experiments*

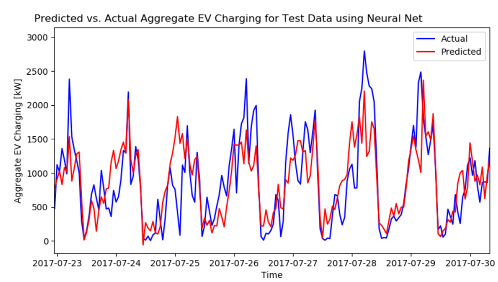


*Fig. 8: Performance of neural network for seven day-ahead forecasts*

## VI. Conclusion

In this study, we used unsupervised clustering techniques to potentially distinguish user-terminated charging behavior and programmable charging equipment. Furthermore, we are able to classify an EV model from its charging history with 78% accuracy. Lastly, we developed a neural network that can forecast day-ahead EV charging load for a residential neighborhood 21% more accurately than the baseline persistence model. The range in typical EV charging load throughout the day is approximately 95% of the peak load. Upon inspection of utility costs and typical transformer layout, we estimate that smart charging of EV loads could save utilities $800 per EV in grid improvement costs through distribution upgrade deferral.

Future work will involve estimating power demand for individual EVs via utilization of an LSTM model. We hope to then quantify load flexibility of EV aggregations, and expand our scope to include ChargePoint data from the entire Bay Area to calculate the broader economic benefit of smart charging.

## VII. Contributions

**Antonio**

- Set up algorithm to scrape data from Pecan Street Dataport
- Pre-processed CSV files for use in main code
- Set up neural network code in PyTorch for load forecasting

**Justin**

- Set up code to load data and extract/compute features and store in dataframe
- Implemented k-means for typical charging session profile and DBSCAN for programmable charger identification
- Implemented grid search over all regression models for load forecasting

**Robert**

- Set up code to iterate through all pre-processed CSV files (multiple households and years) to perform data cleaning
- Completed and tuned PyTorch neural net for load forecasting
- Implemented semi-supervised GMM EM for EV model identification

**Evenly Split**

- Designing and Presenting Poster
- Report Writing

## VIII. Acknowledgments

We would like to acknowledge Dr. Yuting Ji (CEE), Professor Ng, Professor Dror and the CS229 teaching staff for providing mentorship and support.

Code for this study is available via zip-file download from Stanford Google Drive (requires stanford.edu organization access): `https://drive.google.com/file/d/1cex4H1nxwqW7ZtsPUiHhVrQM3a_VVzdp/view?usp=sharing`. Code base is primarily written in Python and Jupyter notebooks. The NumPy library was used for scientific programming, Matplotlib library used for data visualization, Sci-kit learn library used for adapting machine learning models from and performing grid search, and PyTorch library used for creating neural networks [8] [9] [10] [7].

## References

[1] "California will put 5 million electric cars on the road by 2030." `http://www.latimes.com/politics/essential/la-pol-ca-essential-politics-updates-gov\-brown-california-will-put-5-1516999162\-htmlstory.html`. Accessed: 2018-11-18.

[2] M. Kazerooni and N. C. Kar, "Impact analysis of ev battery charging on the power system distribution transformers," in *2012 IEEE International Electric Vehicle Conference*, pp. 1–6, March 2012.

[3] S. G. A. Schuller, C. Flath, "Quantifying load flexibility of electric vehicles for renewable energyintegration," *Applied Energy*, vol. 151, pp. 335–344, 2015.

[4] M. S. N. Sadeghianpourhamami, N. Refa and C. Develder, "Quantitive analysis of electric vehicle flexibility: A data-driven approach," *Electrical Power and Energy Systems*, vol. 95, pp. 451–462, 2018.

[5] "Pecan Street Dataport." `https://dataport.cloud/`. Accessed: 2018-11-18.

[6] D. P. Doane, "Aesthetic frequency classification," *American Statistician*, vol. 30, pp. 181–186, 1976.

[7] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," 2017.

[8] T. Oliphant, "NumPy: A guide to NumPy." USA: Trelgol Publishing, 2006–. [Online; accessed ¡today¿].

[9] J. D. Hunter, "Matplotlib: A 2d graphics environment," *Computing In Science & Engineering*, vol. 9, no. 3, pp. 90–95, 2007.

[10] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.