

# Ensemble machine learning-based algorithm for electric vehicle user behavior prediction<sup>☆,☆☆</sup>

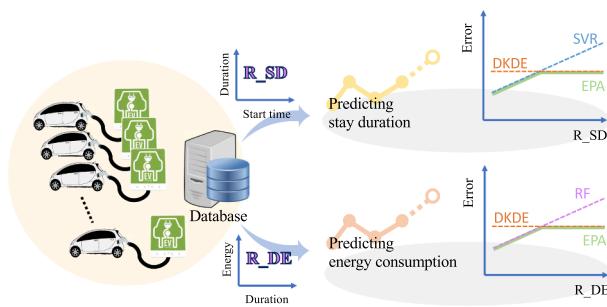
Yu-Wei Chung\*, Behnam Khaki, Tianyi Li, Chicheng Chu, Rajit Gadh

*Smart Grid Energy Research Center (SMERC), University of California, Los Angeles, USA*

## HIGHLIGHTS

- Real electric vehicle charging data from 252 users were analyzed.
- Defining the data entropy/sparsity ratio ( $R$ ) as an indicator for predicting algorithm selection.
- Exploiting the benefit of using diffusion-based kernel density estimator (DKDE) for prediction with high  $R$  data.
- Reducing at least 10% of prediction error compared to a single predicting algorithm.

## GRAPHICAL ABSTRACT



## ARTICLE INFO

### Keywords:

Data entropy  
Data sparsity  
EV user behavior prediction  
Kernel density estimator  
Machine learning

## ABSTRACT

This research investigates electric vehicle (EV) charging behavior and aims to find the best method for its prediction in order to optimize the EV charging schedule. This paper discusses several commonly used machine learning algorithms to predict charging behavior, including stay duration and energy consumption based on historical charging records. It is noted that prediction error increases along with the rise of data entropy or the decrease of data sparsity. Thus, this paper accounts for both indicators by defining the entropy/sparsity ratio ( $R$ ). When  $R$  is low, support vector regression (SVR) and random forest (RF) regression show better accuracy for stay duration and energy consumption predictions, respectively. While  $R$  is high, a diffusion-based kernel density estimator (DKDE) performs better for both predictions. The three methods are assembled as the proposed Ensemble Predicting Algorithm (EPA) to improve predicting performance by decreasing 11% of the duration and 22% of the energy consumption prediction errors. The prediction results are then applied to an optimal EV charging scheduling algorithm to minimize load variance while reducing the EV charging cost. A numerical simulation using real charging data is conducted to show the effectiveness of improved predictions and EV load management. The results show that the charging scheduling combined with EPA prediction can reduce 27% of peak load, 10% of load variation, and 4% cost reduction, compared to uncoordinated charging.

\* This document is a collaborative effort.

\*\* This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

\* Corresponding author at: 44-120 Engr.IV, 420 Westwood Plaza, Los Angeles, CA 90095, USA.

E-mail addresses: [ywchung@ucla.edu](mailto:ywchung@ucla.edu) (Y.-W. Chung), [behnamkhaki@ucla.edu](mailto:behnamkhaki@ucla.edu) (B. Khaki), [tianyi3gli@ucla.edu](mailto:tianyi3gli@ucla.edu) (T. Li), [peterchu@ucla.edu](mailto:peterchu@ucla.edu) (C. Chu), [gadh@ucla.edu](mailto:gadh@ucla.edu) (R. Gadh).

Nomenclature	
$u_{id}$	EV user ID
$t_s$	start charging time [h]
$t_d$	EV stay duration [h]
$\hat{t}_d$	prediction of $t_d$ [h]
$d_w$	day of week
$e$	energy consumption [kWh]
$\hat{e}$	prediction of $e$ [kWh]
$N$	number of charging sessions
$\hat{P}$	prediction value
$T$	actual value for $\hat{P}$
$y$	response variable for regression
$x$	explanatory variable for regression
$\omega$	normal vector for SVR
$\xi$	slack variable in SVR
$\alpha$	Lagrange multiplier in SVR
$h$	bandwidth of the GKDE
$t_{diff}$	diffusion time for DKDE
$N$	total number of EVs
$M$	total number of EV chargers
$z_j$	energy stored in the $j - th$ EV [kWh]
$u_j$	EV charging power [kW]
$d_B$	building load [kW]
$C_j$	the $j - th$ EV battery capacity [kWh]
$g_l$	the $l - th$ EV charger's power rating [kW]
$t$	time index in EV scheduling algorithm [hr]
$H$	prediction horizon in EV scheduling algorithm
$\Omega$	total netload demand [kW]
$\bar{\Omega}$	average netload demand [kW]
$\Pi$	time of use price vector [\$]
$\mathbb{R}$	real Numbers
$\mathbb{X}$	space of input data
$\mathbb{N}$	natural number
$\Phi(\cdot)$	transformation function
$k(\cdot)$	Kernel function in SVR
$K(\cdot)$	Gaussian Kernel function
$P(\cdot)$	probability density function

## 1. Introduction

Electric vehicles (EVs) have received more and more attention since they became an essential part of a smart grid. This is not only because they are environmentally friendly but also because they provide an economical option to people, considering the high price of dwindling fossil fuels. According to *InsideEVs'* statistical report for 2018, around 361 thousand EVs sold in the US while 2 million in total worldwide, and the numbers almost doubled in comparison to that in 2016 [1]. Currently, there are over 614 thousand EVs on the road in California [2], spurred by the government's zero-emission vehicle mandate to achieve the goal of accommodating 1.5 million EVs by 2025 (California Executive Order B-16-2012). Therefore, a sharply increasing number of EVs on the road is foreseen. However, the increasing number of EVs also means that the rise of energy demand and is now becoming a challenge to the electricity grid. Based on the EV charging data collected on the University of California, Los Angeles (UCLA) campus, the average energy consumption is about 8 kWh per charge, which is similar to a daily household energy demand. EV charging load often shows two peaks in a day, one in the morning when people plug in the EV at the workplace and the other in the evening when people get home from work. Without proper energy management for EV charging, the huge power demand due to a large number of plugged-in EVs can stress the distribution grid, degrade the power quality [3,4], and impact the wholesale electricity market [5]. The AAA Foundation report reveals that US drivers spend only 0.8 h in average behind the wheel everyday [6] and mostly leave vehicles parked. This implies that EVs can have a great flexibility for charging and is it not necessary to start charging right after plugged-in. Thus EV charging scheduling plays an important role in distributing and allocating the charging time according to the EVs' availability for overall load management. A proper EV load management not only mitigates the adverse effects of EV charging but also brings benefits to the grid such as load valley filling and peak shaving [7]. Also EV as a mobile battery has a potential to participate in electricity market [8]. Yet, the stochasticity of EV user charging behaviors, including start time, stay duration, and energy demand, poses a significant challenge for the management of charging scheduling. Therefore, this paper discusses and compares several commonly used prediction methods, aiming at developing an accurate predicting model for EV user behavior in order to improve energy management performance. In addition, since the predicting methods, such as regression or kernel density estimator (KDE) are based on the historical data and the historical charging patterns may be very different from each other, there is no one-size-fits-all predicting method for all different EV users. Thus,

this paper analyzes and classifies the different charging patterns, and uses different predicting algorithms accordingly.

Forecasting EV load and its impact to a distribution grid has recently been brought to light by the development of smart grids and the growing number of EVs. However, due to limited access to real EV charging data, synthetic data from travel surveys are used for the majority of these studies. Gennaro et al. [9] utilized the data collected from conventional fuel vehicles. Harris et al. [10] synthesized EV charging profile by using vehicle trip data from the National Household Travel Survey (NHTS). Wang et al. [11] simulated EV energy consumption using car travel survey. In spite of the early stage of EV adoption, some utilities and aggregators have been collecting data from charging stations to gain insight into EV user behavior [12,13]. EA Technology [14] have conducted a three year project to collect data and investigate the impact of clusters of EVs on the electricity grid in the UK. There are two types of data can be used for the forecasting, which are station record and charging record. Station record comes directly from the measurement at the charging outlets while charging record comes from the measurement of each user's charging session. In other words, station record is the aggregated load data over time and charging record is the data for a specific user during a charging session. In [15], multiple methods including a k-nearest neighbor (KNN), a lazy-learning algorithm and a pattern sequence algorithm have been evaluated for aggregated EV load estimation. In [16,17], an autoregressive integrated moving average (ARIMA) method has been proposed for aggregated EV load forecasting. In [18], a data mining model was developed to predict EV charging demand for a geographical area. In [19], modified pattern-based sequence forecasting (MPSF) was in comparison with KNN, support vector regression (SVR) and random forest (RF) algorithms and showed more accurate performance. Also, the aggregated data of EV loads may be used for coordinating the EV charging operation as in [20]. However, to schedule the charging when EVs are plugged in, the charging parameters in each session are preferred instead of the aggregated load information. Furthermore, the prediction by aggregated load requires a large amount of EV charging data and currently the availability of the data is limited. The author in [21] discussed EV charging load forecasting by using station records and charging records, and the results showed that charging record based prediction is faster and more accurate. The method such as Gaussian-based kernel density estimator (GKDE) has been applied to handle the uncertainties of user behaviors for each charging session in [22–24]. But the use of optimal bandwidth selection for GKDE, a.k.a the normal reference rule [25], usually leads to an oversmoothed probability and results in less accurate prediction. To overcome the deficiency of the

normal reference rule, kernel density estimation via diffusion (DKDE) [26], which provides a better bandwidth selection approach, has been used to improve the prediction accuracy of EV charging behavior [27,28]. By examining the performance of the algorithms applied to EV user behavior prediction, it is noted that the variances of the errors are usually very large. This is because the EV charging patterns vary significantly and there is no unique algorithm that works for all. Ref. [28] compares and discusses DKDE and GKDE, and the result shows that DKDE has a higher accuracy for the users who charge their EVs regularly while GKDE works better for the irregulars. However, the overall performance for the prediction still has room for improvement. To the best of the author's knowledge, there is no effective feature that can categorize different EV charging patterns associated with the most accurate predicting algorithms. Therefore, this paper aims at classifying different charging patterns and uses the best approach to predict the charging behavior in each classification.

The rest of the paper is organized as follows: Section 2 describes the method for EV user behavior prediction and reviews commonly used machine learning algorithms. Section 3 presents the EV charging scheduling framework. Section 4 discusses the EV charging datasets and data processing. Section 5 shows the preliminary results and the proposed algorithm. Section 6 presents and discusses the result. Finally, Section 7 concludes the paper.

## 2. User behavior prediction

This section describes the method for EV user behavior prediction. The objective is to predict each specific EV user's stay duration and energy demand based on their historical charging data when they plug in their EVs. For each charging session, a 5-tuple of parameters is used to describe a charging behavior:

$$s \triangleq (u_{id}, t_s, t_d, d_w, e), \quad (1)$$

where  $u_{id}$  is the unique identifier (*user ID*) for each user in our system;  $t_s$  and  $t_d$  denote *start time* and *stay duration*, respectively;  $d_w$  denotes *day of week*; and  $e$  denotes *energy consumption*. Those charging parameters are of vital importance for EV charging scheduling algorithms to determine an optimal solution. To be specific, once a user initiates a charging session, the predictions of *stay duration* and *energy consumption* are required for the scheduling services to determine energy allocation schedule. It is noted that *stay duration* is related to *start time* and *day of week* since users in our model may have their fixed weekly working schedules. Therefore, the prediction of *stay duration* ( $\hat{t}_d$ ) can be expressed as follows:

$$\hat{t}_d = f_d(t_s, d_w). \quad (2)$$

Also, *energy consumption* is related to *start time*, *day of week*, and *stay duration*, such that:

$$\hat{e} = f_e(t_s, d_w, \hat{t}_d). \quad (3)$$

As shown in (3), when predicting *energy consumption*, the *stay duration* is unknown and thus rely on its predicted value,  $\hat{t}_d$ . To gain better prediction accuracy, this paper analyzes several different prediction algorithms aiming at finding the best predictors, i.e.  $f_d$  and  $f_e$ , for each user. The predicting procedure is illustrated in Fig. 1.

Because the charging pattern varies from each other, there is no one-size-fits-all predictor for all EV users. Therefore, in this paper, EV users charging patterns will be classified, and eight different prediction algorithms will be applied to those different classes for comparison to find the optimal solution. To evaluate the performances of different prediction algorithms, symmetric mean absolute percentage error (SMAPE) is chosen here based on the following reasons:

1. SMAPE is a unit free percentage error and it is easier to present the prediction accuracy with different data sets, which are *stay duration*

and *energy consumption* in this paper.

2. Percentage error such as mean absolute percentage error (MAPE, defined as  $MAPE = \text{mean}(|y - \hat{y}|/y)$ ) has a problem when  $y$  value becomes very small. This small value will result in a huge error that bias the overall accuracy. Therefore, SMAPE would be more accurate since it considers both  $y$  and  $\hat{y}$  in the denominator, given that the data is strictly positive.
3. SMAPE is widely used in evaluating EV charging prediction accuracy, it would be easier for comparison.

For charging session  $i$ , the SMAPE is defined as:

$$SMAPE(i) = \frac{1}{N} \sum_{i=1}^N \frac{|\hat{P}(i) - T(i)|}{\hat{P}(i) + T(i)}, \quad (4)$$

where  $N$  is the number of charging sessions,  $\hat{P}$  is the prediction, and  $T$  is the corresponding true value.

### 2.1. Machine learning algorithms

Eight prediction algorithms are reviewed here in the following subsections. By examining the EV charging data, the data can be roughly classified into four categories: linear, non-linear, clustered, and scattered patterns. Multiple linear regression is suitable for a linear pattern. SVR can predict both linear and non-linear patterns and is not biased by outliers. Decision tree (DT), random forest (RF) regression are appropriate for clustered patterns. RF can be more accurate than DT since DT may easily lead to over-fitting. However, it is required to determine the proper number of trees for RF. KNN regression can also be applied for a clustered pattern. Scattered pattern is challenging for prediction. In this case, GKDE and DKDE are used to find the probability density function and make a prediction by calculating the expected value. A statistical method is applied here for comparison. These algorithms are compared, and their effectivenesses are evaluated for different EV charging patterns.

#### 2.1.1. Statistical method

Statistical method such as historical average are referred to as a naive approach, and it is a simple algorithms that used only for comparison with the other forecasting techniques. For the historical average algorithm, the prediction is the average of the past data.

#### 2.1.2. Multiple Linear Regression (MLR)

MLR is used to describe the mathematical relationship between several explanatory variables and a response variable, and the goal is to make predictions about the response variable based on the known explanatory variables according to this relationship. For example, to predict a *stay duration* based on the *start time* and *day of week*. The model of MLR with  $k$  explanatory variables and  $n$  observations is as follows:

$$y_i = b_0 + b_1 x_{i1} + b_2 x_{i2} + \dots + b_k x_{ik} + e_i \text{ for } i = 1, 2, \dots, n, \quad (5)$$

where  $y_i$  is the response variable,  $b_0$  is the y-intercept term,  $[b_1, b_2, \dots, b_k]$  are the regression coefficients,  $[x_{i1}, x_{i2}, \dots, x_{ik}]$  are explanatory variables and  $e_i$  is the error term, which is also known as residual that is used to

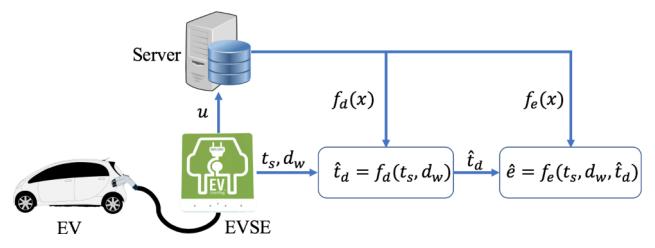


Fig. 1. User behavior prediction.

account for the difference between the actual outcome and the prediction. In this paper, for *stay duration* prediction,  $x_{i1}$  is *start time* and  $x_{i2}$  is *day of week* ( $k = 2$ ). For *energy consumption*,  $x_{i1}$  is *start time*,  $x_{i2}$  is *day of week*, and  $x_{i3}$  is *stay duration* ( $k = 3$ ). Here we use the Python package (sklearn.linear\_model.LinearRegression) [29] for the MLR model.

### 2.1.3. Support Vector Regression (SVR)

SVR is a type of support vector machine that supports linear and non-linear regression. Unlike general linear regression methods, which try to minimize the error between the prediction and data, SVR makes sure the errors do not exceed the threshold. Specifically, in  $\varepsilon$ -SVR [30], the goal is to find a function  $\hat{y}(x)$  that has at most  $\varepsilon$  deviation from the obtained targets  $y_i$  for the training data, ignoring the outliers that locate outside of the  $\varepsilon$ -tolerance band. Consider a training dataset  $\{(x_1, y_1), \dots, (x_n, y_n)\} \subset \mathbb{X}$ , where  $\mathbb{X}$  denotes the space of the input data, SVR can be expressed as follows:

$$\hat{y}(x) = \langle \omega, x \rangle + b \text{ with } \omega \in \mathbb{X}, b \in \mathbb{R}, \quad (6)$$

where  $\omega$  and  $b$  are the solutions of the following optimization problem:

$$\begin{aligned} \min_{w, b, \xi} \quad & \frac{1}{2} \left\| \omega \right\|^2 + C \sum_{i=1}^n \left( \xi_i + \xi_i^* \right) \\ \text{subject to} \quad & \begin{cases} y_i - \langle \omega, x_i \rangle - b \leq \varepsilon + \xi_i \\ \langle \omega, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0. \end{cases} \end{aligned} \quad (7)$$

In Eq. (7), slack variables  $\xi_i, \xi_i^*$  are introduced to handle the problem of infeasible  $\varepsilon$ -precision constraints. The constant  $C > 0$  controls the trade-off between the flatness of  $\hat{y}(x)$  (which is  $\|\omega\|^2$ ) and the number of training data points that deviate larger than  $\varepsilon$  is tolerated. This optimization problem can be solved by Lagrange multipliers method and the solution is given by

$$\omega = \sum_{i=1}^n \left( \alpha_i - \alpha_i^* \right) \Phi(x_i) \text{ and } f(x) = \sum_{i=1}^n \left( \alpha_i - \alpha_i^* \right) k(x_i, x) + b, \quad (8)$$

where  $\alpha_i, \alpha_i^*$  are Lagrange multipliers in which  $\alpha_i, \alpha_i^* \in [0, C]$ ,  $\Phi(x_i)$  is a transformation function, and  $k(x_i, x) := \langle \Phi(x_i), \Phi(x) \rangle$  is a kernel function. The kernel function transforms the data into a higher dimensional feature space to make it possible to perform the linear regression. The Gaussian radial basis function (RBF) is used here as a kernel function:

$$k(x, x_i) = e^{-\frac{\|x-x_i\|^2}{2\sigma^2}}, \quad (9)$$

where  $\sigma \in \mathbb{R}$  is the kernel parameter. The detail of the SVR formulation can be found in [31]. In this paper, we use the Python package (sklearn.svm.svr) [29] for the SVR model.

### 2.1.4. Decision tree (DT) regression

Decision tree (DT) regression is a regression model in the form of a tree structure that breaks down a dataset into smaller classified subsets using each of the independent variables' split points. The average of the classified subset is the prediction value for the target with respect to its corresponding independent variable values. The classified subsets are called leaf nodes whereas the split points are decision nodes. For each decision node, mean square error (MSE) are compared across the independent variables and the variable/point rendering the lowest MSE is chosen as the root node/decision node. The process is recursively continued until the optimal split of the data is achieved, which is defined in terms of tree size constraints within the Python package (sklearn.tree.DecisionTreeRegressor) [29] used in this paper.

### 2.1.5. Random Forest (RF) Regression

Random forest (RF) regression is an ensemble learning method that combines and averages decisions from a sequence of DT models. Formally, RF regression can be expressed as follows:

$$g(x) = \frac{1}{N_{tree}} \sum_{i=1}^{N_{tree}} f_i(x), \quad (10)$$

where  $g(x)$  is the RF model,  $f_i(x)$  is the  $i^{th}$  DT model, and  $N_{tree}$  is the number of decision trees. Each  $f_i(x)$  is built from a sample drawn with replacement from the training dataset. By using the average of the multiple DT models on the corresponding sub-samples of the dataset, the predictive accuracy can be improved.

Here we use the Python package (sklearn.ensemble.RandomForestRegressor) [29] for the RF model.

### 2.1.6. K-Nearest Neighbor (KNN) Regression

K-Nearest Neighbor (KNN) is a non-parametric method used for classification and regression [32]. The regression model is used since the data labels are continuous instead of discrete variables. The model implements learning based on the  $k$ -nearest neighbors of each query point, where  $k = 4$  is specified in this paper. The prediction of a query point is the average of its nearest neighbors, and it is assumed that each neighbor contributes uniformly to the classification of the query point.

Here we use the Python package (sklearn.neighbors.KNeighborsRegressor) [29] for the KNN model.

### 2.1.7. Kernel Density Estimator (KDE)

Kernel density estimator (KDE) is an nonparametric estimation method that does not require explicit parametric model to fit the data. Different from the above-mentioned regression methods, a bi-variate KDE makes use of the joint probability distribution of *start time* vs. *stay duration* and *stay duration* vs. *energy consumption* to predict *stay duration* and *energy consumption*, respectively. To estimate a random variable  $X$ , i.e.  $\hat{x}$ , given a joint probability function  $p(x, y)$ , the expected value is calculated:

$$\hat{x} = E[X] = \int_X x \cdot p(x) dx = \int_X x \cdot \int_Y p(x, y) dy \cdot dx. \quad (11)$$

For example, to predict *stay duration*  $d$  for a specific user, the joint distribution of *start time* ( $t_s$ ) vs. *stay duration* ( $t_d$ ) is used and the univariate distribution for  $d$  is calculated by:

$$p(t_d) = \int_{\Delta t_s} p(t_s, t_d) dt_s = \int_{t_s-\Delta t}^{t_s+\Delta t} p(t_s, t_d) dt_s, \quad (12)$$

where  $\Delta t_s$  is the tolerance interval of start time  $t_s$  and is set to 2 h in this paper from  $t_s - 1$  to  $t_s + 1$  ( $\Delta t = 1$ ). The estimated *stay duration* ( $\hat{t}_d$ ) is the expected value within this interval. Similarly, the energy demand ( $\hat{e}$ ) is estimated by the distribution of *stay duration* ( $t_d$ ) vs. *energy consumption* ( $e$ ) according to the estimated *stay duration* ( $\hat{t}_d$ ) and the tolerance interval ( $\Delta \hat{t}_d$ ) so that:

$$p(e) = \int_{\Delta \hat{t}_d} p(t_d, e) dt_d = \int_{\hat{t}_d-\Delta t}^{\hat{t}_d+\Delta t} p(t_d, e) dt_d \quad (13)$$

In this paper,  $\Delta \hat{t}_d = 2$  where  $\Delta t = 1$ .

**Gaussian-Based Kernel Density Estimator(GKDE)** Given an observed dataset  $X = [X_1, X_2, \dots, X_N]$ , the probability density function can be estimated as follows [33]:

$$\hat{P}_{GKDE}(x) = \frac{1}{Nh} \sum_{i=1}^N K\left(\frac{x - X_i}{h}\right), \quad (14)$$

where  $N$  is the size of  $X$ ,  $h$  is the bandwidth of the Gaussian kernel  $K(\cdot)$  which is defined as:

$$K(u) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{1}{2}u^2\right). \quad (15)$$

Bandwidth  $h$  defines the shape of the kernel function, thus it is a deterministic factor to the performance of the estimator. A large  $h$  oversmoothes the density function that masks the structure of data

while a small  $h$  generates a spiky one that makes the interpretation difficult. It is desired to find a value of  $h$  that minimizes the error between the estimated density and the actual density. However, there is a bias-variance trade-off for the bandwidth selection, which means a large bandwidth reduces the variance of  $\hat{P}_{\text{KDE}}(x)$  but increases the bias with respect to the actual density. On the other hand, a small bandwidth decreases the bias of  $\hat{P}_{\text{KDE}}(x)$  at the expense of larger variance. Silverman's rule of thumb [25], also known as the normal reference rule, provides a simple solution for the optimal bandwidth, with the assumption that the actual density has Gaussian normal distribution. But, this method usually leads to an oversmoothed result in multimodal models such as EV user charging behaviors.

**Diffusion-Based Kernel Density Estimator(DKDE)** Different from the normal reference rule, the optimal bandwidth can be derived from the observed dataset  $X$  using an improved plug-in method introduced in [26].

The KDE of (14) can be expressed in an alternative form:

$$\hat{P}_{\text{KDE}}(x; t_{\text{diff}}) = \frac{1}{N} \sum_{i=1}^N \phi(x, X_i; t_{\text{diff}}), \quad (16)$$

where

$$\phi(x, X_i; t_{\text{diff}}) = \frac{1}{\sqrt{2\pi t_{\text{diff}}}} \exp\left(-\frac{(x - X_i)^2}{2t_{\text{diff}}}\right), \quad (17)$$

in which  $\sqrt{t_{\text{diff}}} = h$  is defined as in (14).

It is interesting to note that GKDE in (16) is the unique solution to the Fourier heat equation as follows [26]:

$$\frac{\partial}{\partial t} \hat{P}(x; t_{\text{diff}}) = \frac{1}{2} \frac{\partial^2}{\partial x^2} \hat{P}(x; t_{\text{diff}}), \quad x \in \mathbb{R}, t_{\text{diff}} > 0, \quad \Delta \equiv \mathbb{R}, \quad (18)$$

with initial condition:

$$\hat{P}(x; 0) = \frac{1}{N} \sum_{i=1}^N \delta(x - X_i), \quad (19)$$

where  $\hat{P}(x; 0)$  represents the empirical density of  $X$ , and  $\delta(x - X_i)$  is the Dirac measure at  $X_i$ .

The Neumann boundary condition is as follows:

$$\frac{\partial}{\partial t} \hat{P}(x; t_{\text{diff}})|_{x=1} = \frac{\partial}{\partial t} \hat{P}(x; t_{\text{diff}})|_{x=0} = 0. \quad (20)$$

Making use of the link between GKDE and Fourier heat equation, finding the optimal bandwidth of (16) is equivalent to finding the optimal mixing time  $t^*$  of the diffusion process governed by (18).

Considering all these conditions and the finite domain  $[0, 1]$ , the analytical solution of (18) is obtained by:

$$\hat{P}_{\text{DKDE}}(x; t_{\text{diff}}^*) = \frac{1}{N} \sum_{i=1}^N \kappa(x, X_i; t_{\text{diff}}^*) x \in [0, 1], \quad (21)$$

in which the kernel function is given by:

$$\kappa(x, X_i; t_{\text{diff}}^*) = \sum_{k=-\infty}^{\infty} \phi(x, 2k + X_i; t_{\text{diff}}^*) + \phi(x, 2k - X_i; t_{\text{diff}}^*) \quad \text{for } x \in [0, 1]. \quad (22)$$

### 3. EV charging scheduling

#### 3.1. Model description

The Electric Vehicle Charging Infrastructure (EVCI) is controlled and managed by a control entity (CE). The purpose of CE is twofold: minimizing the peak load, which is equivalent to load variance minimization, as well as reducing total charging cost [34]. The EVCI is supplied by an electrical feeder shared with an office building which its average net load demand is 318 kW.

Let's denote the total number of EVs and EV chargers (EVCs) by  $N$  and  $M$ , respectively, where  $N, M \in \mathbb{N}$ . In this paper, each EVC has four charging outlets and can charge four EVs at the same time, so we use  $N_l$  to show the set of EVs supplied by  $EVC_l$ . The model of EVCI and the building can be written as:

$$z_j(t+1) = z_j(t) + T_h u_j(t), \quad j = 1, \dots, N; \quad (23a)$$

$$e_{CIB}(t) = d_B(t) + \sum_{j=1}^N u_j(t), \quad (23b)$$

where  $z_j(t) \in \mathbb{X}_j \in \mathbb{R}$ ,  $u_j(t)$ ,  $e_{CIB}(t)$ ,  $d_B(t) \in \mathbb{R}$  and  $t \in \mathbb{N}$ .  $z_j(t)$  in [kWh] is the energy stored in  $EV_j$ ,  $e_{CIB}(t)$  in [kW] is the total net load demand of the EVCI and the building.  $d_B(t)$  in [kW] is the load demand of the building. In (23a),  $T_h$  (in hours [h]) is the discretization in time, e.g.  $T_h = 0.5$  corresponds to 30 min in this paper.  $u_j(t)$ , which is the EV charging power, is introduced as the optimization variable.  $u_j(t)$  is positive in the charging mode and it is negative in discharging mode, a.k.a vehicle to grid (V2G) function. However, V2G is not discussed in this paper.

The constraints on the energy capacity of EVs are:

$$\underline{C}_j(t) \leq z_j(t) \leq \bar{C}_j(t), \quad j = 1, \dots, N, \quad (24)$$

where  $\underline{C}_j(t)$  and  $\bar{C}_j(t)$  are the bounds on the energy stored in EVs. The constraints on charging power of EVs and their corresponding EVCs are:

$$\underline{u}_j \leq u_j(t) \leq \bar{u}_j, \quad j = 1, \dots, N; \quad (25a)$$

$$\underline{g}_l(t) \leq u_l(t) \leq \bar{g}_l(t), \quad l = 1, \dots, M, \quad (25b)$$

where  $\underline{u}_j$  and  $\bar{u}_j$  are the minimum and maximum power ratings of the EV charging outlet, and  $\underline{g}_l$  and  $\bar{g}_l$  are the minimum and maximum power ratings of the EVCs.

The bounds in (24) for EVs are time-varying and defined as follows; if  $EV_j$ ,  $j = 1, \dots, N$  is:

- not plugged in EVC,  $\underline{C}_j(t) = \bar{C}_j(t) = 0$
- plugged in EVC, but it is in idle mode,  $\underline{C}_j(t) = 0$  &  $\bar{C}_j(t) = C_j$
- plugged in EVC, and it is needed by time  $t$ ,  $\underline{C}_j(t) = \bar{C}_j(t) = C_j$

where  $C_j$  is the maximum capacity of  $EV_j$ 's battery.

#### 3.2. Problem formulation

For the given time index  $t$  and prediction horizon  $H \in \mathbb{N}$ , let's denote the vector notation  $\mathbf{u}_j = (u_j(t), u_j(t+1), \dots, u_j(t+H-1))^T$ ,  $\mathbf{u}_j(t) \in \mathbb{R}^H$ , which is used for all other variables as well. To formulate the objective function, we define total net load demand at time  $t$  by (26):

$$\Omega := \mathbf{d}_B. \quad (26)$$

Also, the average net load demand at time  $t$  over time horizon  $H \in \mathbb{N}$  is:

$$\bar{\Omega} := \frac{1}{H} \sum_{t=k}^{k+H-1} \Omega. \quad (27)$$

Accordingly, the twofold objective function of EV coordinated charging (CC) is written as:

$$V := \min_{\mathbf{u}} \sum_{j=1}^N \beta (\Pi^T \mathbf{u}_j) + \left( \bar{\Omega} - \left( \Omega + \sum_{j=1}^N \mathbf{u}_j \right) \right)^2 \text{ s. t. } (23)-(25), \quad (28)$$

where  $\beta$  is a weighting factor, and  $\Pi \in \mathbb{R}^H$  is time of use (TOU) price vector [35]. The first part in (28) reduces charging cost, while the second part minimizes the total load variance.

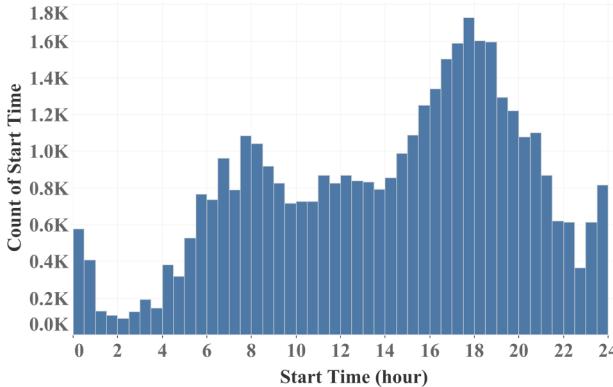


Fig. 2. Statistics of EV charging start time.

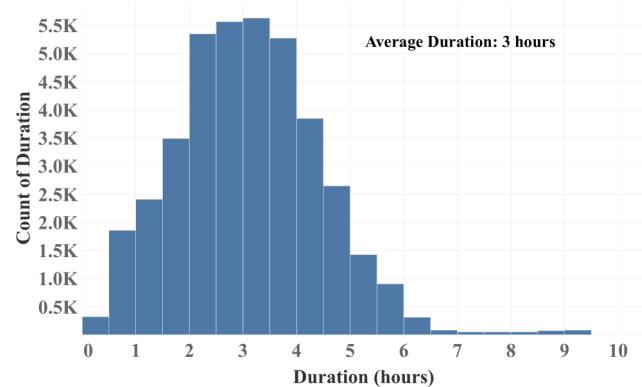


Fig. 3. Statistics of EV stay duration.

#### 4. Data preparation

##### 4.1. Data

Two sources of EV charging data were applied to this research, including SMERC charging stations on the UCLA campus [12] as working space and real residential EV users' data in the UK that is available from the EA technology website [14]. The data used from the UCLA charging stations was recorded from October 1, 2015 to December 31, 2017 and the data from the EV technology between February 16, 2014 and November 29, 2015. However, not every user in those datasets has a charging history that is long enough for data analysis and prediction. Therefore, we selected 50 users' data from UCLA and 202 from EV technology, which have at least 100 charging records, with 39,458 records in total. The data was split into 70% for the training set, 20% for the validation set, and 10% for the test set.

The statistics for charging *start time*, *stay duration*, and *energy consumption* are shown in the figures below. Fig. 2 shows two peaks for EV charging *start time*: one at 7:30 in the morning and the other at 17:30 in the evening. The average *stay duration* is 3 h, and the average *energy consumption* is 10.63 kWh, as shown in Figs. 3 and 4, respectively.

##### 4.2. Data preprocessing

The charging *start time* and *stay duration* were converted to hour. For instance, 13:15 will be noted as 13.25 h. If a *stay duration* was smaller than 0.5 h or an *energy consumption* was smaller than 1 kWh, the entire 5-tuple parameter for that charging session was removed from the dataset. Also, if an *energy consumption* was mistakenly recorded as more than the physical maximum of the charging device, the record value was replaced by the maximum value of its historical *energy consumption*.

##### 4.3. Data entropy

Joint entropy is used here to characterize the uncertainty of a set of variables. Two kinds of datasets were analyzed, which are *start time* vs. *duration* data and *duration* vs. *energy consumption* data. For calculation, *start time* and *duration* are rounded to the closest half hour, and the *energy consumption* is rounded to the closest integer. The values of *start time* and *duration* are then mapped into a set of integers  $\in [0, 47]$ , which represents [0: 00, 23: 30]. The formulation of a joint entropy is as follows:

$$E(X, Y) = - \sum_{x \in X} \sum_{y \in Y} P(x, y) \log_2 \left[ P(x, y) \right], \quad (29)$$

where  $x$  and  $y$  are the two variables in dataset  $X$  and  $Y$ , respectively;  $P(x, y)$  is the joint probability of the two variables.

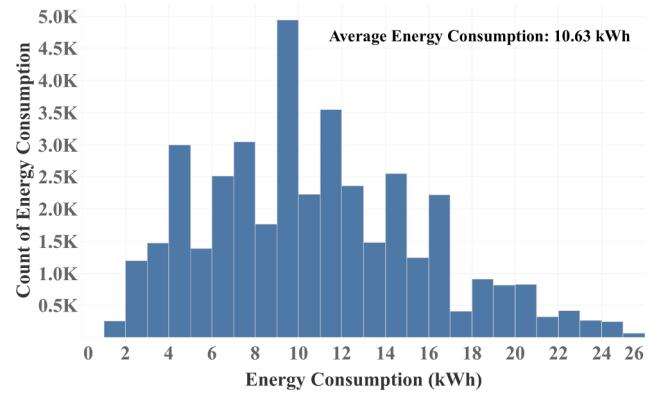


Fig. 4. Statistics of EV energy consumption per charge.

##### 4.4. Data sparsity

Sparsity is defined as the number of zero entries divided by the total number of entries. Intuitively, if a sparsity is high the data is less variant because most entries are repeated. On the other hand, for low sparsity, the data is more scattered. As was the data entropy discussed in the previous section, *start time* vs. *duration* data and *duration* vs. *energy consumption* data are analyzed. The values of *start time*, *duration*, and *energy consumption* are rounded. Following are the examples of sparsity calculation: In Fig. 5, the numbers in the cells are the number counts for the data points. For *start time* vs. *duration*, the *start time* ranges from 0 to 23 while the *duration* from 0 to 9. The number of non-zero entries is 31 and the total entries is 240, thus the sparsity is  $(240-31)/240 = 0.87$ . In the same manner, the sparsity for *duration* vs. *energy consumption* is 0.89.

#### 5. Preliminary results and proposed algorithm

##### 5.1. Preliminary results

Figs. 6 and 7 show the comparisons of eight algorithms' prediction errors with regard to data entropy, data sparsity and the ratio of entropy/sparsity ( $R$ ). Generally, SMAPE positively correlates to data entropy and negatively correlates to data sparsity. Therefore, this paper takes into account both of the effects of entropy and sparsity by defining the ratio:  $R = \text{entropy}/\text{sparsity}$ . SD and DE represent the datasets of *Start time* vs. *Duration* and *Duration* vs. *Energy Consumption*, respectively. Dataset SD is used to predict *stay duration* while DE is utilized to predict *energy consumption*. Ratios of  $R_{SD}$  and  $R_{DE}$  are calculated using the training datasets' data entropy and sparsity.  $\rho$  is the correlation coefficient of SMAPE and  $R$ .  $P$ -value indicates the statistical significance of the trend (significant if  $P$ -value  $< 0.05$ ).

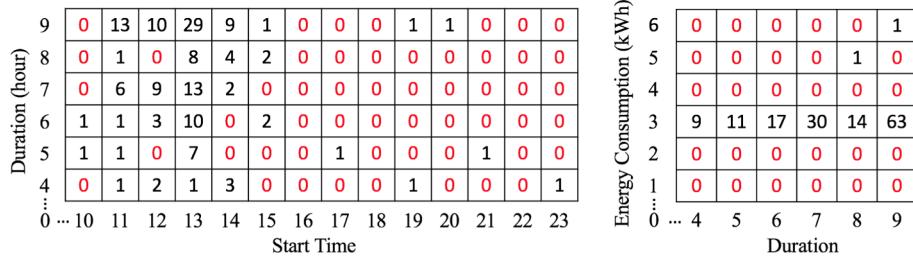


Fig. 5. Sparsity of EV charging patterns. left: start time vs. duration, right: duration vs. energy consumption.

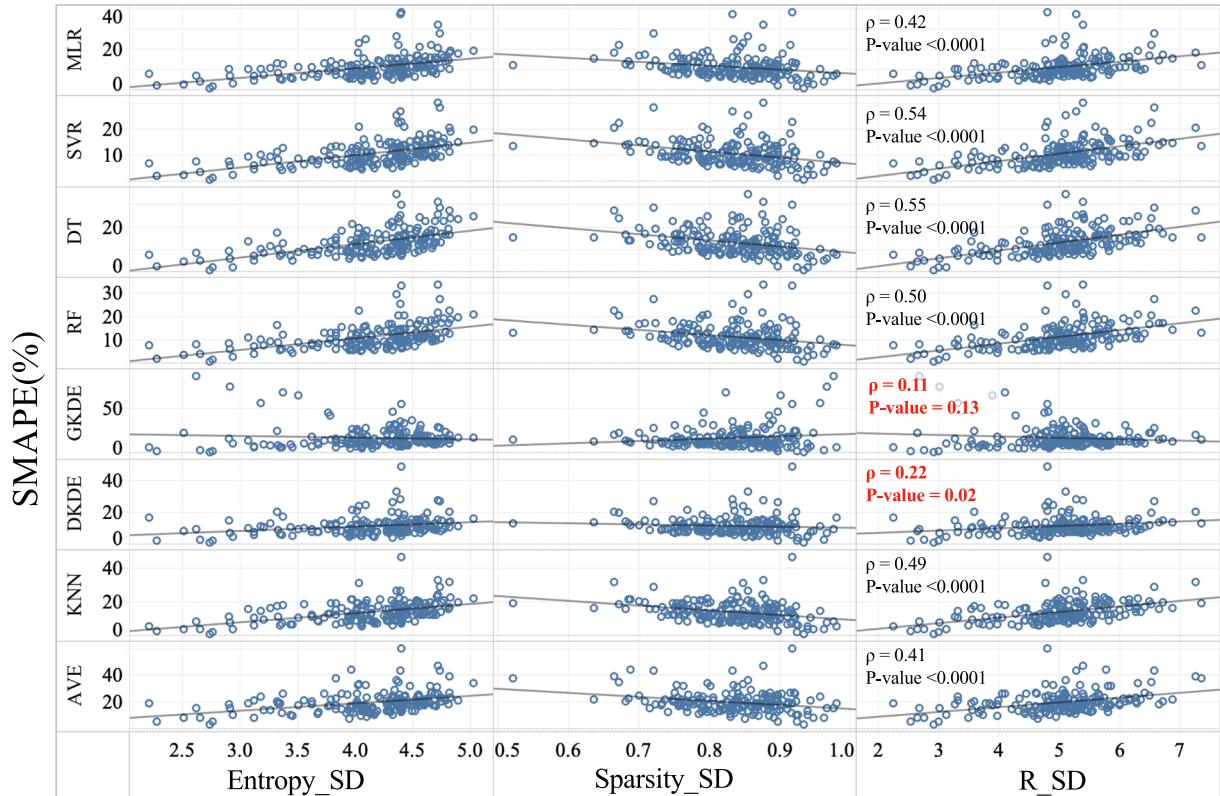


Fig. 6. Comparisons of SMAPE(%) versus entropy, sparsity and R\_SD (entropy/sparsity).

The SMAPEs of MLR, SVR, DT, RF, and KNN are compared with KDE as shown in Figs. 8 and 9. Fig. 8 compares the SMAPEs of *duration*, and it shows that when  $R_{SD}$  is larger than 5.5, DKDE performs better. Likewise, Fig. 9 compares the SMAPEs of *energy consumption* predictions, and it shows that DKDE performs better when  $R_{DE}$  is larger than 4.

Table 1 shows the *duration* prediction results of the different algorithms. It indicates that SVR is most accurate overall, especially when  $R_{SD} \leq 5.5$ . DKDE is the best when  $R_{SD} > 5.5$  and the SMAPE does not change significantly in different  $R_{SD}$  categories.

Table 2 shows the *energy consumption* prediction results of the different algorithms. RF is shown to be the most accurate overall, especially when  $R_{DE} \leq 4$ . Similarly, DKDE performs the best when  $R_{DE} > 4$ .

## 5.2. Proposed algorithm

Based on the preliminary results, the combination of SVR, RF, and DKDE is proposed to form an ensemble algorithm, namely the EPA. The algorithm is depicted in Fig. 10 below.

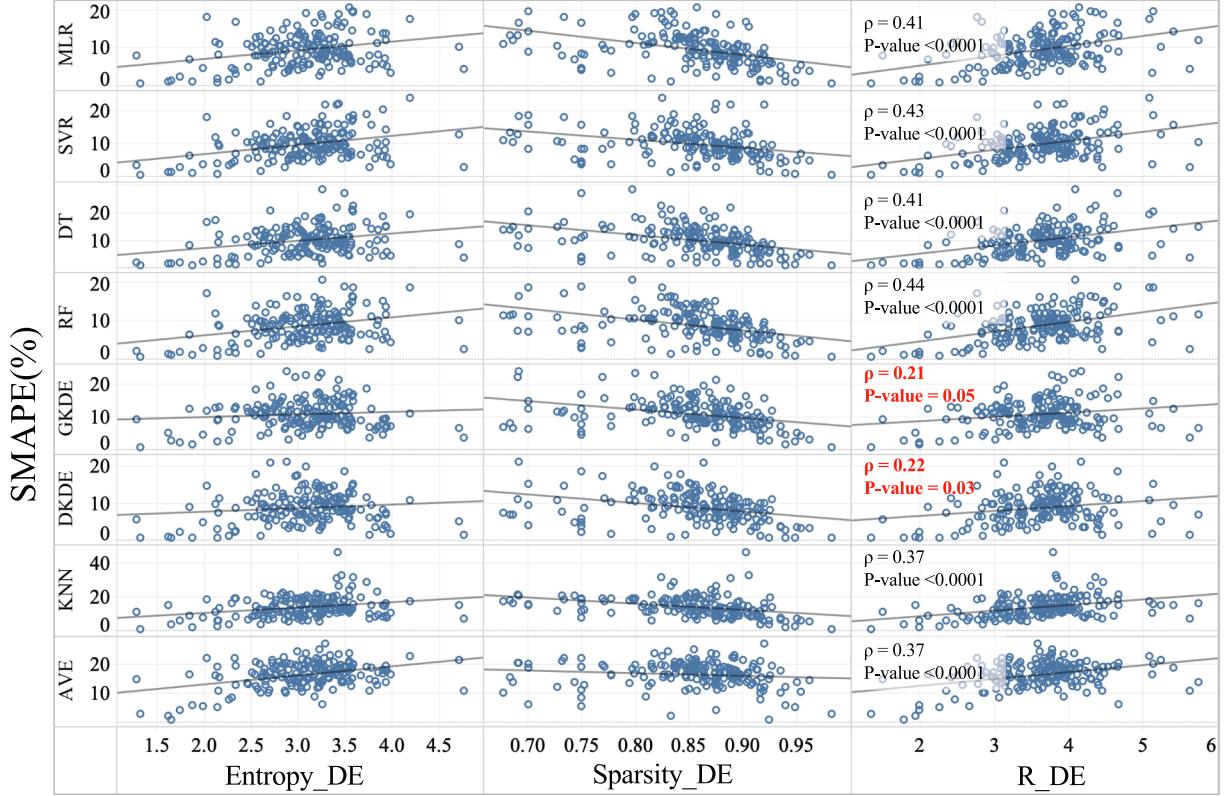
Data entropy and sparsity are analyzed for all registered EV users in the system in order to calculate the R value. When an EV is plugged in,

the user's  $R_{SD}$  is retrieved to determine either SVR or DKDE to be used for predicting *stay duration*. The predicted stay duration is then sent to the next step for *energy consumption* prediction. Similarly, RF or DKDE is applied depending on the value of  $R_{DE}$ . The threshold of R to switch the algorithms may need to update quarterly since user behaviors may change over time. The EPA is evaluated using a 10% test dataset. The prediction results along with the EV scheduling results are presented in the next section.

## 6. Results and discussion

Figs. 11 and 12 show the SMAPE with regard to R for *duration* and *energy consumption* predictions, respectively. Fig. 11 illustrates the comparison between SVR and DKDE. As shown in the figure, the SMAPE of SVR is smaller when  $R_{SD}$  is smaller than 5.5, whereas the SMAPE of DKDE is smaller when  $R_{SD}$  is larger than 5.5. Fig. 12 demonstrates the comparison between RF and DKDE. As expected, RF is more accurate when  $R_{DE}$  is smaller than 4, while DKDE performs better when  $R_{DE}$  is larger than 4.

Table 3 shows the average and standard deviation for the SMAPE of *duration* and *energy consumption* predictions. A pairwise T-test with the null hypothesis that the EPA has the same performance as the other



**Fig. 7.** Comparisons of SMAPE(%) versus entropy, sparsity and R\_DE (entropy/sparsity).

algorithms is rejected by the small P-values ( $p < 0.05$ ). The results show that EPA has decreased the errors significantly for *duration* and *energy consumption* predictions by around 11% and 22%, respectively.

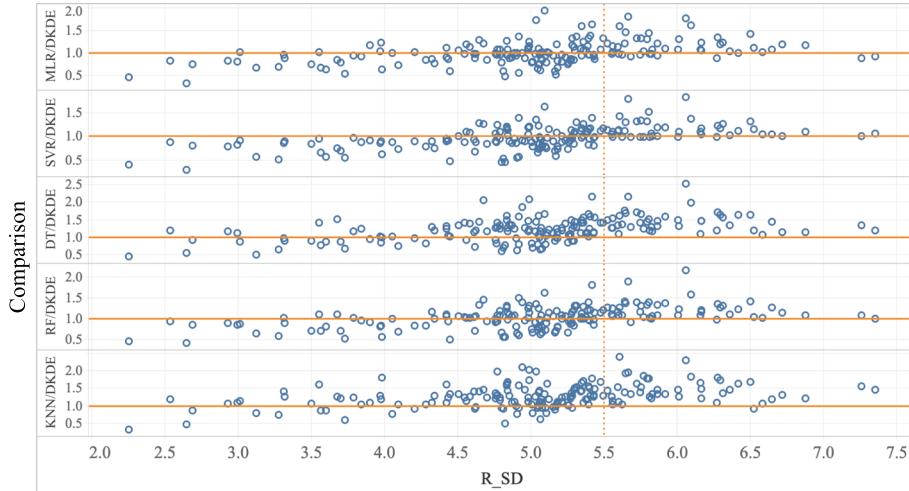
Root mean squared error (RMSE) is also evaluated to show the effectiveness of EPA. Since each user has different number of charging records, here we calculate the mean of RMSE of all users, called mean estimation deviation (MED). MED is defined as follows:

$$MED = \frac{1}{N_{user}} \sum_{i=1}^{N_{user}} \left( \sqrt{\frac{1}{N_i} \sum_{j=1}^{N_i} (\hat{P}(j) - T(j))^2} \right), \quad (30)$$

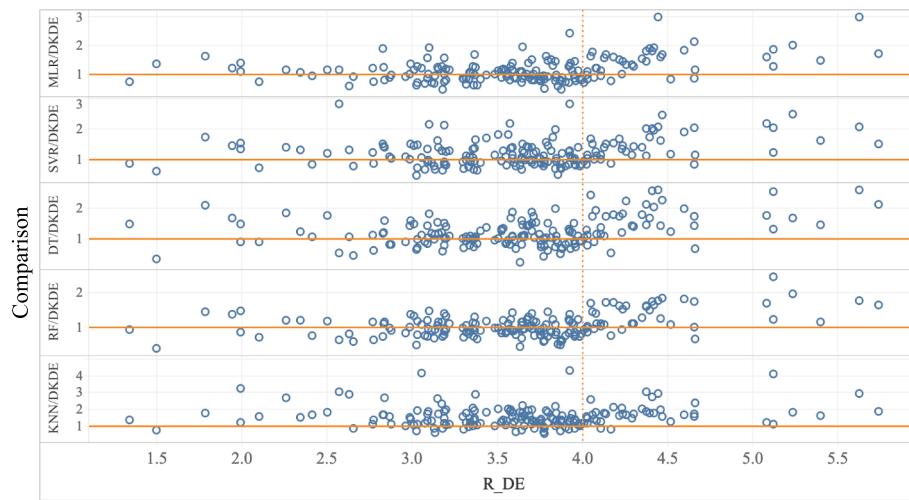
where  $N_{user}$  is the number of users,  $N_i$  is the number of charging sessions for the  $i$ -th user,  $\hat{P}$  is the prediction, and  $T$  is the corresponding true

value. **Table 4** shows the comparison of MED among SVR, DKDE, RF, and EPA. For EPA, the prediction MED for *stay duration* is 1.16 h and for *energy consumption* is 2.52 kWh.

Using EPA prediction results, we run CC for the EVCI including 252 EVs and 63 EVCs. TOU used in our numerical simulation is shown in Fig. 13. Total charging profiles of the EVCI for uncoordinated charging (uCC) and CC using real and prediction data are shown in Fig. 14. As it is clear, CC flattens the total load profile which results in peak load shaving when TOU price is high and valley filling when TOU price is low. Also, the difference between load profile using EPA and real data is negligible during most of the time intervals. However, there appears a valley during 23:00 and 4:00 in Fig. 14 and is not filled. This is because of less availability of EVs according to our dataset, in which the “end



**Fig. 8.** Comparisons of different algorithms with DKDE for duration prediction.



**Fig. 9.** Comparisons of different algorithms with DKDE for energy consumption prediction.

**Table 1**

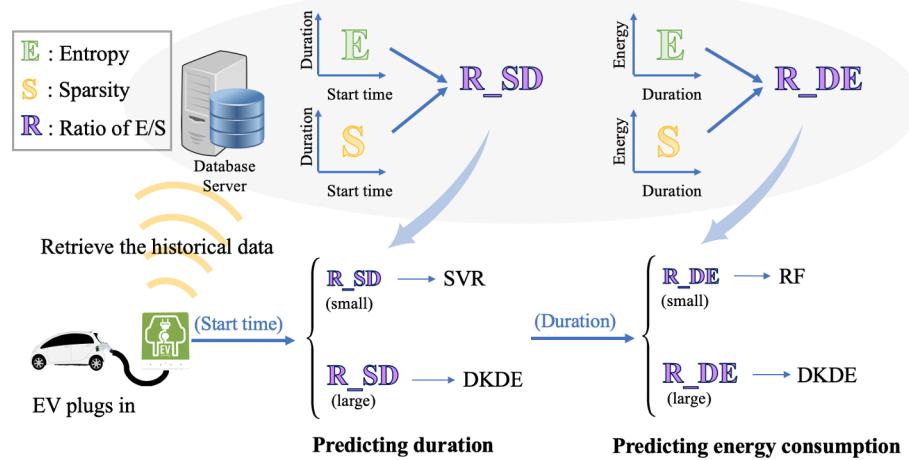
Average and Standard deviation (in parentheses) for the SMAPE(%) of duration prediction.

Ratio (R_SD)	SVR	MLR	DT	RF	DKDE	GKDE	KNN	AVE
R_SD<5.5 (n = 187)	9.54 (4.66)	10.19 (5.91)	11.84 (5.62)	10.21 (5.32)	10.96 (6.37)	17.39 (14.32)	12.65 (6.26)	18.43 (7.78)
R_SD>5.5 (n = 65)	13.40 (4.40)	13.51 (4.20)	16.66 (4.78)	14.15 (4.33)	11.81 (4.15)	14.90 (5.84)	16.82 (4.95)	23.47 (7.61)
Overall (n = 252)	10.54 (4.89)	11.05 (5.69)	13.09 (5.80)	11.23 (5.36)	11.18 (5.88)	16.75 (12.72)	13.73 (6.21)	19.73 (8.03)

**Table 2**

Average and Standard deviation (in parentheses) for the SMAPE(%) of energy consumption prediction.

Ratio (R_DE)	SVR	MLR	DT	RF	DKDE	GKDE	KNN	AVE
R_DE <= 4 (n = 204)	9.06 (4.25)	8.71 (4.14)	9.16 (4.55)	7.96 (3.68)	8.31 (4.20)	10.33 (4.23)	11.80 (5.38)	16.10 (4.64)
R_DE > 4 (n = 48)	12.91 (4.11)	12.69 (4.32)	13.68 (5.52)	11.59 (4.04)	10.54 (3.76)	12.68 (4.80)	15.70 (5.54)	17.86 (3.21)
Overall (n = 252)	9.79 (4.48)	9.46 (4.45)	10.01 (5.05)	8.65 (4.00)	8.73 (4.21)	10.78 (4.43)	12.54 (5.61)	16.43 (4.45)



**Fig. 10.** Flowchart of the ensemble predicting algorithm.

time” refers to the “end charging time” instead of “un-plugging time,” and therefore further restrains the EVs’ time flexibility for charging.

The coordinated and uncoordinated EV charging scheduling results with respect to peak-to-peak (PTP) and root-mean-square (RMS) of the load profile, and charging cost are shown in Table 5. The result of CC\_EPA aligns well with CC\_Real, and reduces 27% peak load, 10% load variation, and 4% charging cost from that of uCC’s.

Although the result shows only \$10 can be saved per day by scheduling 252 EVs’ charging comparing to uncoordinated charging, with a large number of EVs, the saving can be significant. Furthermore, according to [36], the energy unit cost (EUC) negatively related to a load factor (LF) along with a hyperbolic function ( $EUC \propto 1/LF$ ), where LF is defined as follows:

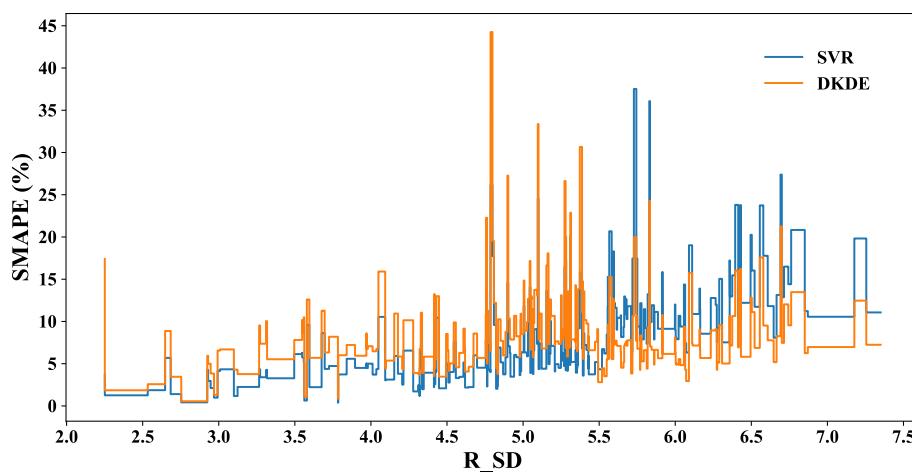


Fig. 11. Average SMAPE vs. R\_SD for SVR and DKDE.

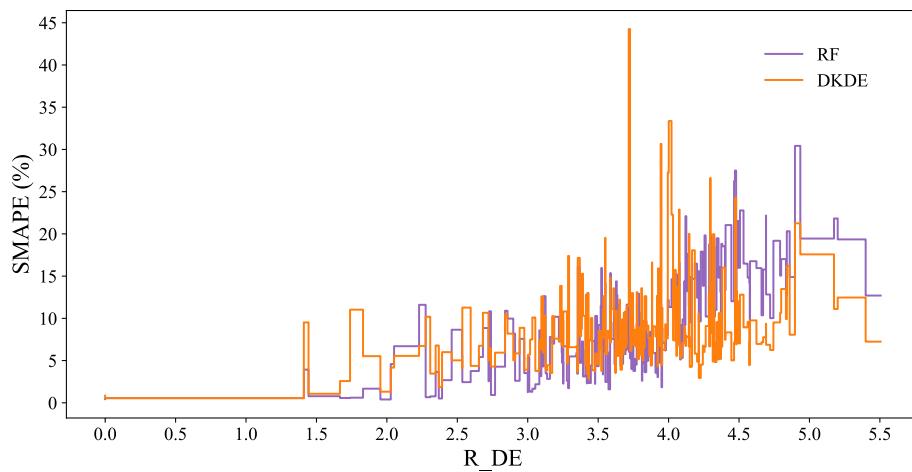


Fig. 12. Average SMAPE vs. R\_DE for RF and DKDE.

**Table 3**

SMAPE(%), standard deviation (in parentheses), and the pairwise T-test result.

Algorithm	Duration prediction			Energy consumption prediction		
	SVR	DKDE	EPA	RF	DKDE	EPA
SMAPE (%)	11.53 (5.18)	11.67 (6.36)	10.40 (5.80)	9.69 (4.60)	9.56 (4.26)	7.54 (4.24)
P-value	0.00964409	0.00833339	-	0.00182736	$2.00649 \times 10^{-5}$	-

**Table 4**

MED (Duration: hour; Energy: kWh), standard deviation (in parentheses), and the pairwise T-test result.

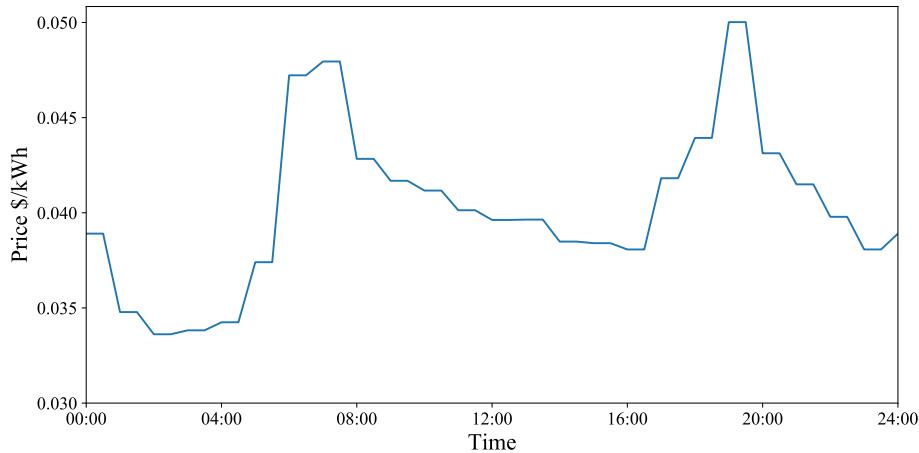
Algorithm	Duration prediction			Energy consumption prediction		
	SVR	DKDE	EPA	RF	DKDE	EPA
MED	1.36 (0.69)	1.38 (0.47)	1.16 (0.54)	2.94 (1.35)	2.65 (0.87)	2.52 (0.97)
P-value	0.00449328	$1.3247 \times 10^{-7}$	-	$1.49621 \times 10^{-6}$	0.0017243	-

$$LF = \frac{\text{AveragePower}}{\text{PeakPower}} * 100\%. \quad (31)$$

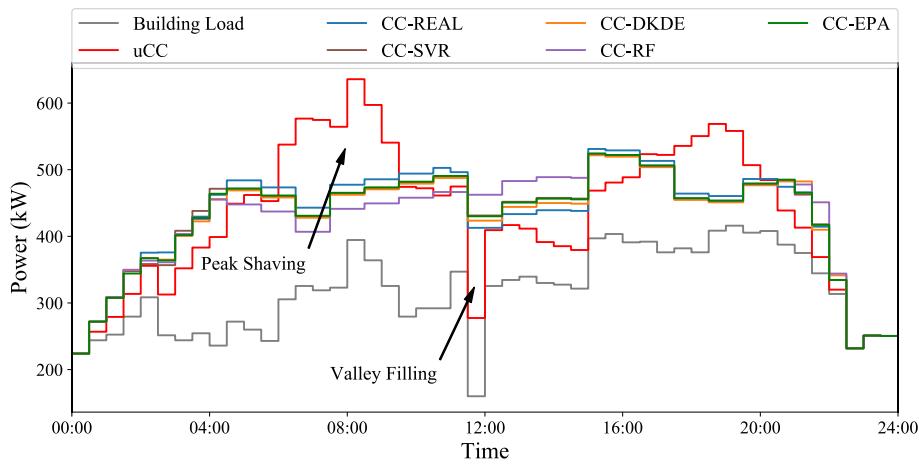
The price will approach the minimum when LF close to 1. As shown in Fig. 2, people tend to plug in EVs in the morning when they arrive at work, and in the evening when they get home. If the energy peak produced by EV charging that drastically lower the LF, the energy price

will increase sharply. Therefore, EV charging control is necessary to accommodate such larger amount of EVs within the electricity grid.

The results show that the EPA model fits the true densities, including start time, stay duration, and energy consumption, better than the other algorithms. Therefore, the control entity (CE) can schedule EV charging optimally in terms of minimizing peak loads and reducing



**Fig. 13.** TOU price for the EV charging scheduling simulation [35].



**Fig. 14.** Load profile using uCC and CC algorithms based on real data.

**Table 5**  
Comparison between uCC with CC using real data and EPA.

Algorithm	PTP (kW)	RMS (kW)	Total Cost (\$)
CC_Real	307.13	139.59	219.94
CC_EPA	300.50	139.24	219.25
uCC	411.83	156.44	229.13

charging cost. For scale-up, a considerable amount of EVs can be utilized to mitigate the renewable energy intermittency issue such as solar duck-curve problem. Since the EPA algorithm can predict the EVs' availability very well, in combination with vehicle-to-grid (V2G) technology, the charging CE can manage to charge EVs during the midday when solar power is ample and discharge during the evening to reduce the peak load. To control a large amount of EVs requires a distributed EV charging scheduling method. However, this method is beyond the discussion here and is elaborated in [34,37].

## 7. Conclusion

This paper develops an improved algorithm for EV user behavior prediction, namely the EPA. It is founded that, in general, predicting SMAPEs positively correlate to data sparsity/entropy ratio ( $R$ ) but this relationship for GKDE and DKDE is relatively weak. Therefore, the KDE method can be utilized to handle the high  $R$  data with lower prediction

error. Based on this property and the analysis result, SVR, RF, and DKDE are selected to compose the EPA. The synergy of the three algorithms enhances the prediction performance where SVR is good at predicting EV stay duration, RF performs better on energy consumption estimation, and DKDE takes care of the prediction with the high  $R$  data. The estimations by EPA are then applied to the optimal EV charging scheduling algorithm for load variation and charging cost minimization. Owing to the increased accuracy of the prediction, the scheduling algorithm can provide better EV charging load management in terms of reducing load variation and charging cost. Real data is employed for a numerical simulation to demonstrate the improved prediction accuracy of EPA and validate the effectiveness of EV charging scheduling algorithm.

The proposed EPA algorithm can be applied to any scale of charging station, with an assumption that EVs' charging records are known. However, to reach optimal scheduling within a distribution grid, the connection between each charging station is required. This can be achieved by adopting the Open Charge Point Protocol (OCPP), which is already available on the market [38]. The data entropy and sparsity are the data property that can be applied to other datasets. This paper presents a preliminary result for how the different prediction algorithms relate to the property  $R$ . To apply the EPA to a different dataset, there is an advantage to use DKDE for high  $R$  data prediction since DKDE is not sensitive to the change of  $R$ . As for the methods for low  $R$  data, it would be different depending on the datasets.

## Appendix A. Supplementary material

Supplementary data associated with this article can be found, in the online version, at <https://doi.org/10.1016/j.apenergy.2019.113732>.

## References

- [1] Loveday S. Plug-In electric vehicle sales report card; June 2019 [accessed: 07.25.2019] <<https://insideevs.com/monthly-plug-in-sales-scorecard/>>.
- [2] Veloz. Sales dashboard. [accessed: 07.25.2019] <<https://www.veloz.org/sales-dashboard/>>.
- [3] Mu Y, Wu J, Jenkins N, Jia H, Wang C. A spatial-temporal model for grid impact analysis of plug-in electric vehicles. *Appl Energy* 2014;114:456–65. <https://doi.org/10.1016/j.apenergy.2013.10.006>.
- [4] Salah F, Ilg JP, Flath CM, Basse H, van Dinther C. Impact of electric vehicles on distribution substations: a Swiss case study. *Appl Energy* 2015;137:88–96. <https://doi.org/10.1016/j.apenergy.2014.09.091>.
- [5] Foley A, Tytler B, Calhan P, Gallachoir B. Impacts of electric vehicle charging under electricity market operations. *Appl Energy* 2013;101:93–102. <https://doi.org/10.1016/j.apenergy.2012.06.052>.
- [6] Johnson T. Americans spend an average of 17,600 minutes driving each year [accessed: 11.10.2018] <<https://newsroom.aaa.com/tag/american-driving-survey/>>.
- [7] Gan L, Topcu U, Low S. Optimal decentralized protocol for electric vehicle charging. *IEEE Trans Power Syst* 2013;28(2):940–51. <https://doi.org/10.1109/TPWRS.2012.2210288>.
- [8] Kristoffersen TK, Capion K, Meibom P. Optimal charging of electric drive vehicles in a market environment. *Appl Energy* 2011;88(5):1940–8. <https://doi.org/10.1016/j.apenergy.2010.12.015>.
- [9] Gennaro MD, Paffumi E, Scholz H, Martini G. GIS-driven analysis of e-mobility in urban areas: an evaluation of the impact on the electric energy grid. *Appl Energy* 2014;124:94–116. <https://doi.org/10.1016/j.apenergy.2014.03.003>.
- [10] Harris CB, Webber ME. An empirically-validated methodology to simulate electricity demand for electric vehicle charging. *Appl Energy* 2014;126:172–81. <https://doi.org/10.1016/j.apenergy.2014.03.078>.
- [11] Wang H, Zhang X, Ouyang M. Energy consumption of electric vehicles based on real-world driving patterns: a case study of Beijing. *Appl Energy* 2015;157:710–9. <https://doi.org/10.1016/j.apenergy.2015.05.057>.
- [12] Smart Grid Energy Research Center (SMERC), UCLA, Smart grid project - smart EV charging station [accessed: 03.01.2018] <<https://evsmartplug.net/smartergrid/ChargingRecord/>>.
- [13] Xiong Y, Wang B, Cheng Chu C, Gadh R. Vehicle grid integration for demand response with mixture user model and decentralized optimization. *Appl Energy* 2018;231:481–93. <https://doi.org/10.1016/j.apenergy.2018.09.139>.
- [14] Eatechnology.com, My electric avenue data [online]; 2016 [accessed: 03.01.2018] URL <<https://www.eatechnology.com/americas/projects/my-electric-avenue/>>.
- [15] Majidpour M, Qiu C, Chu P, Gadh R, Pota HR. Fast prediction for sparse time series: demand forecast of EV charging stations for cell phone applications. *IEEE Trans Ind Inform* 2015;140(1):242–50. <https://doi.org/10.1016/j.epsr.2016.06.003>.
- [16] Amini MH, Kargarian A, Karabasoglu O. ARIMA-based decoupled time series forecasting of electric vehicle charging demand for stochastic power system operation. *Electric Power Syst Res* 2016;111:378–90. <https://doi.org/10.1016/j.epsr.2016.06.003>.
- [17] Wang B, Hu B, Qiu C, Chu P, Gadh R. EV charging algorithm implementation with user price preference. IEEE power energy society innovative smart grid technologies conference (ISGT), 2015 2015. p. 1–5. <https://doi.org/10.1109/ISGT.2015.7131895>.
- [18] Ydadas E, Marmaras C, Cipcigan LM, Jenkins N, Carroll S, Barker M. A data-driven approach for characterising the charging demand of electric vehicles: a UK case study. *Appl Energy* 2016;162:763–71. <https://doi.org/10.1016/j.apenergy.2015.10.151>.
- [19] Majidpour M, Qiu C, Chu P, Gadh R, Pota HR. A novel forecasting algorithm for electric vehicle charging stations. International conference on connected vehicles and expo (ICCVE), 2014 2014. p. 1035–40. <https://doi.org/10.1109/ICCVE.2014.7297504>.
- [20] Xu Z, Hu Z, Song Y, Zhao W, Zhang Y. Coordination of PEVs charging across multiple aggregators. *Appl Energy* 2014;136:582–9. <https://doi.org/10.1016/j.apenergy.2014.08.116>.
- [21] Majidpour M, Qiu C, Chu P, Pota HR, Gadh R. Forecasting the EV charging load based on customer profile or station measurement? *Appl Energy* 2016;163:134–41. <https://doi.org/10.1016/j.apenergy.2015.10.184>.
- [22] Wang B, Wang Y, Nazaripouya H, Qiu C, Chu C, Gadh R. Predictive scheduling framework for electric vehicles with uncertainties of user behaviors. *IEEE Internet Things J* 2017;4(1):52–63. <https://doi.org/10.1109/JIOT.2016.2617314>.
- [23] Wang Y, Shi W, Wang B, Chu C-C, Gadh R. Optimal operation of stationary and mobile batteries in distribution grids. *Appl Energy* 2017;190:1289–301. <https://doi.org/10.1016/j.apenergy.2016.12.139>.
- [24] Wang B, Huang R, Wang Y, Nazaripouya H, Qiu C, Chu C, et al. Predictive scheduling for Electric Vehicles considering uncertainty of load and user behaviors. 2016 IEEE/PES transmission and distribution conference and exposition (T&D 2016). 2016. p. 1–5.
- [25] Silverman BW. *Density estimation for statistics and data analysis*. New York: Routledge; 1998.
- [26] Botvov ZI, Grotowski JF, Kroese DP. Kernel density estimation via diffusion. *Ann Statist* 2010;38(5):2916–57. <https://doi.org/10.1214/10-AOS799>.
- [27] Khaki B, Chung YW, Chu C, Gadh R. Nonparametric user behavior prediction for distributed EV charging scheduling. In: 2018 IEEE power and energy society general meeting conf (PESGM 2018); 2018.
- [28] Chung YW, Khaki B, Chu C, Gadh R. Electric vehicle user behavior prediction using hybrid kernel density estimator. In: 2018 IEEE international conference on probabilistic methods applied to power systems (PMAPS 2018); 2018. p. 1–6.
- [29] Buitinck L, Louppe G, Blondel M, Pedregosa F, Mueller A, Grisel O, et al. API design for machine learning software: experiences from the scikit-learn project. In: ECML PKDD workshop: languages for data mining and machine learning; 2013. p. 108–22.
- [30] Vapnik V. *The nature of statistical learning theory*. Springer; 1995.
- [31] Smola AJ, Schölkopf B. A tutorial on support vector regression. *Tech rep, Statistics and Computing*; 2003.
- [32] Altman NS. An introduction to kernel and nearest-neighbor nonparametric regression. *Am Stat* 1992;46(3):175–85. <https://doi.org/10.1080/00031305.1992.10475879>.
- [33] Cristopher MB. *Pattern recognition and machine learning*. Springer-Verlag; 2016.
- [34] Khaki B, Chu C, Gadh R. Hierarchical distributed framework for ev charging scheduling using exchange problem. *Appl Energy* 2019;241:461–71. <https://doi.org/10.1016/j.apenergy.2019.03.008>.
- [35] California Independent System Operator (CAISO). Locational marginal price [Online]; 2018 [accessed: 01.05.2018] <<http://oasis.caiso.com>>.
- [36] Keyhani A. *Design of smart power grid renewable energy systems*. 2nd ed. Wiley-IEEE Press; 2016.
- [37] Khaki B, Chung YW, Chu C, Gadh R. Hierarchical distributed EV charging scheduling in distribution grids. In: 2019 IEEE power and energy society general meeting conf (PESGM 2019); 2019.
- [38] Open Charge Alliance, Global Platform for Open Protocols. [accessed: 07.25.2019] <<https://www.openchargealliance.org>>.