Classification of Electric Vehicle Charging Time Series with Selective Clustering

Chenxi Sun*‡, Tongxin Li†‡, Steven H. Low† and Victor O.K. Li*
* Department of Electrical and Electronic Engineering
The University of Hong Kong
{cxsun, vli}@eee.hku.hk
† Department of Computing and Mathematical Sciences
California Institute of Technology
{tongxin, slow}@caltech.edu

***Abstract*—We develop a novel iterative clustering method for classifying time series of EV charging rates based on their "tail features". Our method first extracts tails from a diversity of charging time series that have different lengths, contain missing data, and are distorted by scheduling algorithms and measurement noise. The charging tails are then clustered into a small number of types whose representatives are then used to improve tail extraction. This process iterates until it converges. We apply our method to ACN-Data, a fine-grained EV charging dataset recently made publicly available, to illustrate its effectiveness and potential applications.**

***Index Terms*—Time series clustering, EV charging curves**

I. Introduction

According to the International Energy Agency (IEA) [1], the global electric vehicle (EV) stock will exceed 130 million vehicles by 2030. This trend has motivated a large body of EV research in the last decade, from pilot studies to testbeds and data analytics, from charging algorithms to user behavior to optimal investments, from impact on electric grid to energy services. In this paper we develop a method to learn battery behavior based on fine-grained charging data from the field.

There are a large number of battery models for hybrid and electric vehicles in the literature, *e.g.*, those that capture the underlying electro-chemical processes or equivalent circuit behavior [2], [3]. Most of these models however are too detailed for system-level applications such as grid impact study or optimal scheduling. Instead, models that describe the charging/discharging behavior as a battery fills up/depletes will be more suitable. This paper develops a key component of such a model that can be used for real-time optimization or simulation.

Our model is based on a dynamic EV charging dataset, called ACN-Data, collected from smart EV charging facilities, called adaptive charging networks (ACN), at Caltech, JPL, and a Bay Area workplace. See [4], [5] for details of Caltech ACN which has been charging EVs since February 2016 and has delivered close to 850 MWh by October 2019, equivalent to more than 2.5 million vehicle miles traveled and 880 metric tons of avoided greenhouse gases. ACN-Data has recently been released [6] and, to our knowledge, is the only large-scale fine-grained charging data that is publicly available.

Our contributions. This paper presents the first analysis of the charging curves in ACN-Data and develops a systematic method to learn battery behavior from the data. Our main results are two-fold:

1) We develop a novel *iterative clustering framework*, summarized in Fig. 3, for classifying time series of battery charging rates based on their "tail features" (see Definition II.1). A key challenge that our method overcomes is to extract tails from a diversity of charging curves that have different lengths, contain missing data, and are distorted by scheduling algorithms and measurement noise. The charging tails are then clustered into several types whose representatives are then used to improve tail extraction. This process iterates until it converges. To compare charging tails of different lengths, we introduce a modified Euclidean distance function in (5) that achieves a higher silhouette coefficient and better classification accuracy than traditional distance functions (see Section IV-A).
2) We validate our method on ACN-Data. Our result shows that even though the number of charging curves is large, they can be classified into a small number of types (in our experiment, 304 charging curves are classified into 6 types). These battery types can be used to predict charging behavior with good accuracy (generally high R^2 values; see Table II in Section IV-B).

These preliminary results open up venues for future work and potential applications (see Section V). For instance, the representative charging curves from the classification can serve as a building block for online optimization of EV charging and online detection of abnormal battery conditions.

The rest of the paper is organized as follows. We formulate our problem in Section II. We develop our iterative clustering method in Section III. We apply our method to ACN-Data in Section IV and illustrate its effectiveness with an example ap-

This work is supported by NSF through grants CCF 1637598, ECCS 1619352, ECCS 1931662, CPS ECCS 1739355, CPS ECCS 1932611.
‡ Both authors contributed equally to this work.

TABLE I
LIST OF KEY NOTATION USED IN THE PAPER.

Used ACN-Data Fields	
connectionTime	Time when the user plugs in
doneChargingTime	Time of the last non-zero charging current
disconnectTime	Time when the user unplugs
pilotSignal	Time series of pilot signals
chargingCurrent	Time series of actual charging currents
userID	Unique identifier of the user
Clustering Parameters	
$\mathcal{N}$	Set of n charging sessions
$\mathcal{T}$	Set of T charging time slots
$\mathcal{S}$	Set of n charging curves
$\mathcal{C}_k$	Cluster indexed by k ($k = 1, \ldots, K$)
Sequences	
$\mathbf{p}_i$	Pilot curve for session $i \in \mathcal{N}$
$\mathbf{s}_i$	Charging curve for session $i \in \mathcal{N}$
$\mathbf{x}_i$	Charging tail for session $i \in \mathcal{N}$
$\mathbf{c}_k$	Tail representative for cluster $\mathcal{C}_k$

plication. We conclude the paper in Section V with limitations of this work, as well as potential extensions and applications.

II. PROBLEM FORMULATION

A. ACN-Data

An ACN typically consists of tens of level-2 chargers controlled by a local controller that communicates wirelessly with these chargers and servers in the cloud. An ACN is capable of real-time measurement, communication, computing and control. It adapts EV charging currents to driver needs as well as capacity limits of the electric system. A typical charging session starts when a driver plugs in her EV and informs ACN through a mobile app the amount of energy required (in terms of miles) and her estimated departure time. The EV will be charged until either the requested energy is delivered, or the battery is fully charged, or the EV is unplugged, whichever occurs first. The charging currents of all EVs that have not finished charging are jointly optimized and updated every minute. Every 5 to 10 seconds, a control (pilot) signal is sent to the EV and the actual charging current drawn by the vehicle is measured. ACN-Data contains both session data (user's ID, arrival time, departure time, requested energy, and actual energy delivered) and fine-grained charging data at seconds resolution (time series of control signals and charging currents). Unfortunately, the current EV charging standard does not collect batteries' states of charge nor EV specifications. Table I summarizes some of the available features of ACN-Data used in this work. Note that not all sessions contain user inputs (*i.e.*, the last three fields of Table 1 in [6].) In this paper we shall focus on the claimed sessions that are associated with user inputs.

B. Charging curves

With the terminology introduced in Table I, denote by $\mathcal{N} := \{1, \ldots, n\}$ the set of charging sessions. Each charging session refers to the charging duration from *connectionTime* to *disconnectTime* (see Table I). Without loss of generality,

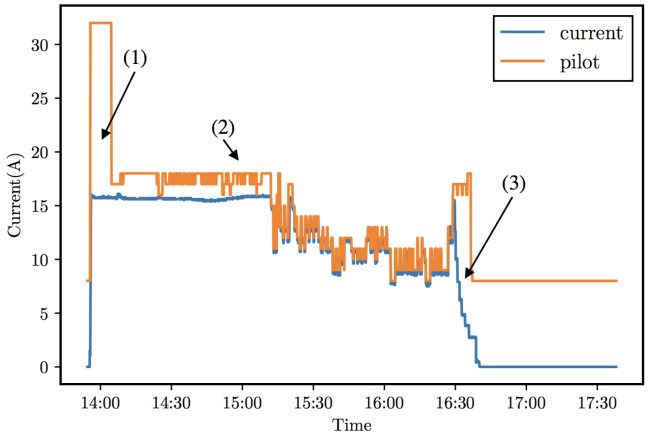

Fig. 1. An example of a charging curve (in blue) and the corresponding pilot curve (in orange) for a charging session with *userID* 409 on Oct. 13, 2018.

we assume the times series of charging currents have the same length T and time granularity (If not, we preprocess the time series as explained in Section III-A and pad the shorter ones with zeros). Let $\mathcal{T} := \{1, \ldots, T\}$ be the set of time slots from *connectionTime* to *disconnectTime*. In the remaining contexts, we refer to "time series" as the raw data and "charging curves" the sequences with equally sampled points after preprocessing (introduced in Section III-A), unless otherwise stated. We first define a charging curve and its associated pilot curve. For any session $i \in \mathcal{N}$, a *charging curve* $\mathbf{s}_i \in \mathbb{R}^T$ is the sequence of actual charging currents during the session i, *i.e.*, $\mathbf{s}_i := (s_i(1), \ldots, s_i(T))$. For any session $i \in \mathcal{N}$, a *pilot curve* $\mathbf{p}_i \in \mathbb{R}^T$ is the sequence of control signals during the session i, *i.e.*, $\mathbf{p}_i := (p_i(1), \ldots, p_i(T))$. At each time $t \in \mathcal{T}$, a charger sends a pilot signal $p_i(t)$ to the vehicle which then draws a current $s_i(t)$ that is no higher than $p_i(t)$ (both $s_i(t)$ and $p_i(t)$ are in units of Amp). Given a set of n charging curves $\mathcal{S} := \{\mathbf{s}_i \in \mathbb{R}^T : i \in \mathcal{N}\}$ and the associated pilot curves $\mathcal{P} := \{\mathbf{p}_i \in \mathbb{R}^T : i \in \mathcal{N}\}$, the key issue considered in this paper is: how to classify the elements of $\mathcal{S}$ into different groups and implement the classification efficiently?

Typically, a charging curve from a charging session consists of two stages – the *bulk charging stage* and the *absorption stage*. In the bulk stage which usually occurs before the state of charge (SoC) reaches 80% full, the charging current is usually equal approximately to the pilot signal and the charging voltage steadily increases. In the absorption stage, the voltage stays approximately at its peak level and the charging currents decreases as the battery reaches full charge. In cases when the available time for charging is sufficiently long, a charging session may contain an additional stage, namely the *idle stage* where the charging current is closed to zero (neglecting noise). An example of a charging curve and its associated pilot curve is shown in Fig. 1. It can be observed that the measured charging current does not follow the pilot signal exactly. The gap between the pilot signal and charging current fluctuates due to the following reasons: (1) the maximum charging current that the vehicle can draw being smaller than the control signal;

(2) random noise; (3) entry into the absorption or idle stage.

C. Charging tails

The charging current in the bulk charging stage is controlled by the scheduling algorithm and therefore it exhibits little information of the battery. For classification purposes, we are mainly interested in the second stage, during which the charging current might exhibit distinct patterns because of different types of batteries. Let t_s^i and t_e^i denote the start time and end time of the absorption stage for session $i \in \mathcal{N}$. We refer to the subsequence of the charging curve in this stage as charging tail, defined as follows.

Definition II.1 (Charging Tail)**.** For session $i \in \mathcal{N}$, a *charging tail* $\mathbf{x}_i := \left(s_i(t), t = t_{\mathrm{s}}^i, \ldots, t_{\mathrm{e}}^i\right)$ is the subsequence of the charging curve $\mathbf{s}_i$ in the absorption stage $\left\{t_{\mathrm{s}}^i, \ldots, t_{\mathrm{e}}^i\right\}$.

Since charging tails display distinctive characteristics of their corresponding charging curves, we will classify charging curves based on their tails. A common battery model assumes that a charging curve starts and stays at some maximum charging current C_{max}^i until the battery enters the absorption stage when the charging current steadily decreases to zero. In this model, the start time t_s^i of the charging tail is easily identifiable to be the last time the charging current stays at the maximum rate C_{max}^i and the end time t_e^i is the first time the charging current drops to zero, *i.e.*, an (ideal) charging tail $\mathbf{x}_i$ is a decreasing sequence defined by: $C_{\mathrm{max}}^i = s(t_s^i) > s(t_s^i + 1) \geq \cdots \geq s(t_e^i - 1) > s(t_e^i) = 0$. In practice, however, extracting the charging tail $\mathbf{x}_i$ from a real charging curve $\mathbf{s}_i$, *i.e.*, identifying the start time t_s^i and end time t_e^i of the absorption stage, can be difficult. A charging curve $\mathbf{s}_i$ is rarely a decreasing sequence as the simple model above assumes. The charging current fluctuates for multiple reasons, not only the *internal* charging state of a battery, but also *external* factors such as pilot signal control (scheduling) or noise. In Fig. 2, we display examples of charging curves where the rates drop due to these reasons.

The confusion caused by scheduling can be cleared up using the first tail extraction method in Section III-B. The confusion caused by noise is trickier to deal with since, in particular, the noise can be large and fluctuate frequently as shown in Fig. 1 and Fig. 2. Thus, it is nontrivial to differentiate the changes due to noise from the other scenarios. In addition, it is possible that more than one scenarios occur simultaneously, *e.g.,*, scheduling within the tail stage. In this case, the charging tails may not be decreasing sequences. Therefore, determining the exact starting point (and ending point) of the absorption stage is difficult. Moreover, for a given length-T charging curve $\mathbf{s}_i \in \mathbb{R}^T$, different tail extraction methods (as introduced in Section III-B) may give distinct tails. Therefore, we consider the set of all candidates of charging tails for session $i \in \mathcal{N}$, denoted by $\mathcal{X}_i$. As subsequences of $\mathbf{s}_i$, the tails in $\mathcal{X}_i$ may not have the same dimension. This motivates a novel *selective clustering* problem with a *new* objective: *How to cluster n candidates (of charging tails) $\{\mathbf{x}_i \in \mathcal{X}_i : i \in \mathcal{N}\}$ with the ability of choosing a candidate $\mathbf{x}_i \in \mathcal{X}_i$ for each charging curve $\mathbf{s}_i$?* In the sequel, we formalize our clustering problem.

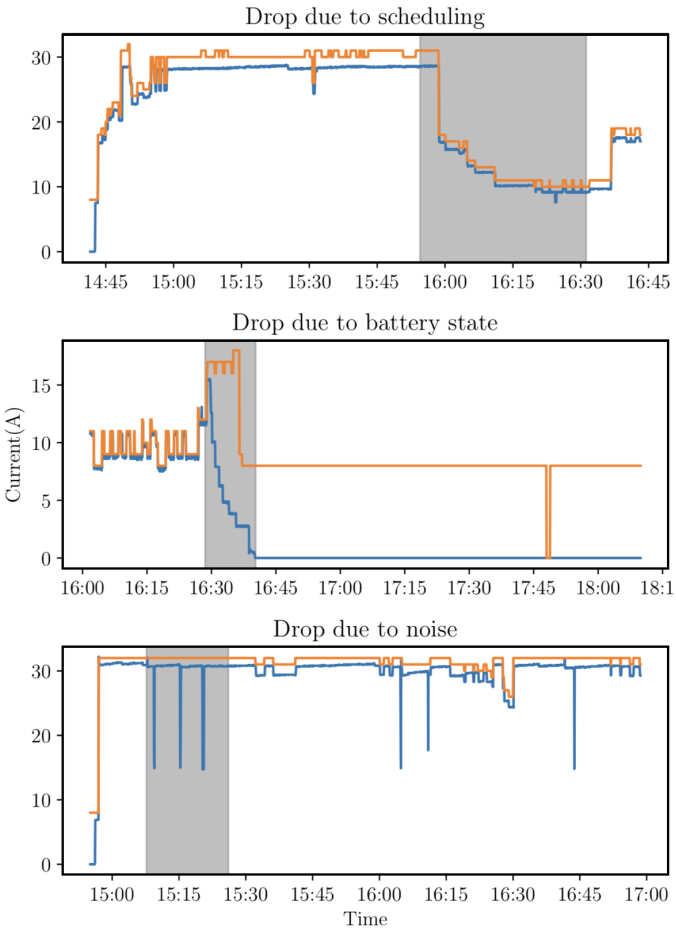

Fig. 2. Examples of charging curves where charging currents drop due to (1) scheduling, (2) battery charging state, and (3) noise, as indicated by the shaded regions. Each plot only shows a selected portion of a session. The time series are for sessions with *userID* 576 (top), 409 (mid) and 526 (bot), obtained on Nov. 07, 2018, Oct. 09, 2018 and Oct. 22, 2018, respectively.

D. Selective clustering

With the above definitions, the charging tail classification problem can be defined as the following optimization:

$$\min_{\mathcal{X}} \min_{\mathcal{C}} \sum_{k=1}^{K} \sum_{i \in \mathcal{C}_k} d(\mathbf{c}_k, \mathbf{x}_i) \tag{1}$$

where $\mathcal{X} := \{\mathbf{x}_i \in \mathcal{X}_i : i \in \mathcal{N}\}$ is a set of n *candidates*, constructed by selecting exactly one tail from $\mathcal{X}_1, \ldots, \mathcal{X}_n$. We assume the number of clusters K is known and searching the best K is beyond the scope of this paper. Let $\mathcal{K} := \{1, \ldots, K\}$. The set $\mathcal{C} := \{\mathcal{C}_k : k \in \mathcal{K}\}$ specifies a partition $\mathcal{N} = \bigcup_{k=1}^{K} \mathcal{C}_k$ of the charging sessions $\mathcal{N}$ with each $\mathcal{C}_k$ representing a distinctive cluster. Moreover, $\mathbf{c}_k$ is a *tail representative* for the k-th cluster, defined as its *medoid*. The *distance function(s)* is denoted by $d(\cdot, \cdot)$, which will be specified in Section III.

To solve the minimization in (1), we use the idea of alternating minimization (AM) and refine the representative of each cluster by iteratively implementing the following until convergence. With suitable initialization, the iterations (the $(\ell + 1)$-step) consist of two main steps.

- *Tail Extraction (TE)*: Given n fixed tail representatives, we find new candidates that minimize the following:

$$\mathbf{x}_i^{(\ell+1)} := \arg\min_{\mathbf{x}\in\mathcal{X}_i} \min_{k\in\mathcal{K}} d\left(\mathbf{c}_k^{(\ell)}, \mathbf{x}\right), \ i \in \mathcal{N}. \quad (2)$$

- *Tail Clustering (TC)*: We cluster the new tails obtained via TE and find new representatives $\mathbf{c}_1^{(\ell+1)}, \ldots, \mathbf{c}_K^{(\ell+1)}$ by solving the following minimization:

$$\min_{\mathcal{C}} \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{C}_k} d\Big(\mathbf{c}_k^{(\ell+1)}, \mathbf{x}_i^{(\ell+1)}\Big). \quad (3)$$

In Algorithm 1, we summarize the iterative process. The details of the initialization step is described in Section III-B. Note that conducting TC and TE repeatedly cannot increase the objective function in (1). Therefore, the AM we established is guaranteed to have local convergence.

Algorithm 1: AM for Selective Clustering

Input: Charging curves $\mathcal{S}$ and pilot curves $\mathcal{P}$;
Output: Clustering $\mathcal{C}$ and representatives $\widehat{\mathbf{c}}_1, \ldots, \widehat{\mathbf{c}}_K$;
$\ell \longleftarrow 1$;
Initialization $\longrightarrow \mathbf{x}_1^{(\ell)}, \ldots, \mathbf{x}_n^{(\ell)}$;
while *not converge* **do**
  Tail clustering (TC) $\longrightarrow \mathbf{c}_1^{(\ell)}, \ldots, \mathbf{c}_n^{(\ell)}$;
  Tail extraction (TE) $\longrightarrow \mathbf{x}_1^{(\ell+1)}, \ldots, \mathbf{x}_n^{(\ell+1)}$;
  $\ell \longleftarrow \ell + 1$
end

Theorem II.1. *With arbitrary initialization $\mathbf{x}_1^{(1)}, \ldots, \mathbf{x}_n^{(1)}$, by iteratively performing (2) and (3), Algorithm 1 converges to a local optimum consisting of representative tails $\widehat{\mathbf{c}}_1, \ldots, \widehat{\mathbf{c}}_K$.*

Proof. First, TE cannot decrease the objective function in (1). For any session $i \in \mathcal{N}$ and pair of $\mathbf{x}_i^{(\ell)} \in \mathcal{C}_{k(i)}$ and the corresponding tail representative $\mathbf{c}_{k(i)}^{(\ell)}$, the minimization in (2) guarantees that there exists a tail representative $\mathbf{c}_{k'}$ such that

$$d\left(\mathbf{c}_{k'}, \mathbf{x}_i^{(\ell+1)}\right) \le d\left(\mathbf{c}_{k(i)}^{(\ell)}, \mathbf{x}_i^{(\ell)}\right).$$

Therefore, this specifies a clustering with the objective function less than or equal to the previous clustering. Similarly, TC cannot decrease the objective function, since we just show that there exists a better clustering for the new tails, and the minimization in (3) can only result in an objective value that is equal to or smaller than the original one. □

The computational complexity for solving (2) in TE is $O(nK\gamma(d))$ where $\gamma(d)$ is the complexity for computing the distance function with fixed input sequences. Our experiments use an approximation in (4) for a more efficient implementation. In practice, for efficiently implementing TC, heuristics are used for finding a local optimum of (3). Moreover, as the AM procedure also leads to a local minimum, an initialization that is close to a global minimum is important. In the next section, we introduce tail extraction methods for initialization and heuristic algorithms for clustering.

III. Classification method

In this section, we present our framework for charging curve clustering. It consists of three main stages depicted in Fig. 3.

A. Preprocessing

In general, the charging curve $\mathbf{s}_i$ and the pilot curve $\mathbf{p}_i$ for session $i \in \mathcal{N}$ are neither sampled at a fixed rate nor perfectly aligned. Most analysis techniques, however, require that the time series be unevenly spaced. We therefore re-sample the time series as the mean over a fixed interval δ (if there is at least one sample) and fill in the missing points by linear interpolation. This preprocessing step ensures the alignment of signals in the time domain $\mathcal{T} = \{1, \ldots, T\}$ so that the distance metric $d(\mathbf{c}, \mathbf{x})$ is well defined.

B. Tail extraction

Not all charging curves contain tails, for two reasons. First, certain batteries do not exhibit a smooth absorption stage and the current just drops from C to 0 directly. Second, EVs may be unplugged before they are fully charged. This can happen when the EV leaves earlier than the input departure time, or when the requested energy is lower than the battery's remaining capacity. We consider three rules of thumb for tail extraction.

1) Extraction by pilot signals: As mentioned in Section II-C, a tail is typically a decreasing sequence. Therefore, we declare that a battery has entered the absorption stage if the charging current $s(t)$ falls below a certain value $C > 0$. The end of the stage is the time when the charging current first reaches approximately zero. This simple rule of thumb is straightforward to implement. A drawback however is that it is hard to determine a suitable threshold $C > 0$. Scheduling, system congestion, or noise may cause the charging current to drop below the threshold C even before entering the absorption stage. To mitigate the confusion due to scheduling, we utilize the pilot curves and call a subsequence $\mathbf{s}'$ of a charging curve $\mathbf{s}$ *piloted at time* t if $p(t) - p(t-1) \ge s(t) - s(t-1)$. We accept a tail if it is not piloted everywhere and $s(t) \le C$ for a given *threshold parameter* $C > 0$.

2) Extraction by duration: The end of the absorption stage can be found by locating the first (approximately) zero value of the charging currents. If we have an estimate of the duration of the adsorption stage, we will be able to extract the tail. This approach requires the knowledge of the tail duration. Moreover, even for the same battery, the duration of the adsorption stage varies across different sessions because of noise and our re-sampling.

The first two extraction methods can be combined to extract tails. In our experiments reported in Section IV, for each distinct user, we regard the first two methods as a two-layer filter and extract a tail representative for each session if the tail passes the filtering criteria. In particular, for session i, we employ grid search for the selection of threshold parameter $C > 0$ by decreasing it from the maximal charging current $C_{\max}^i$.

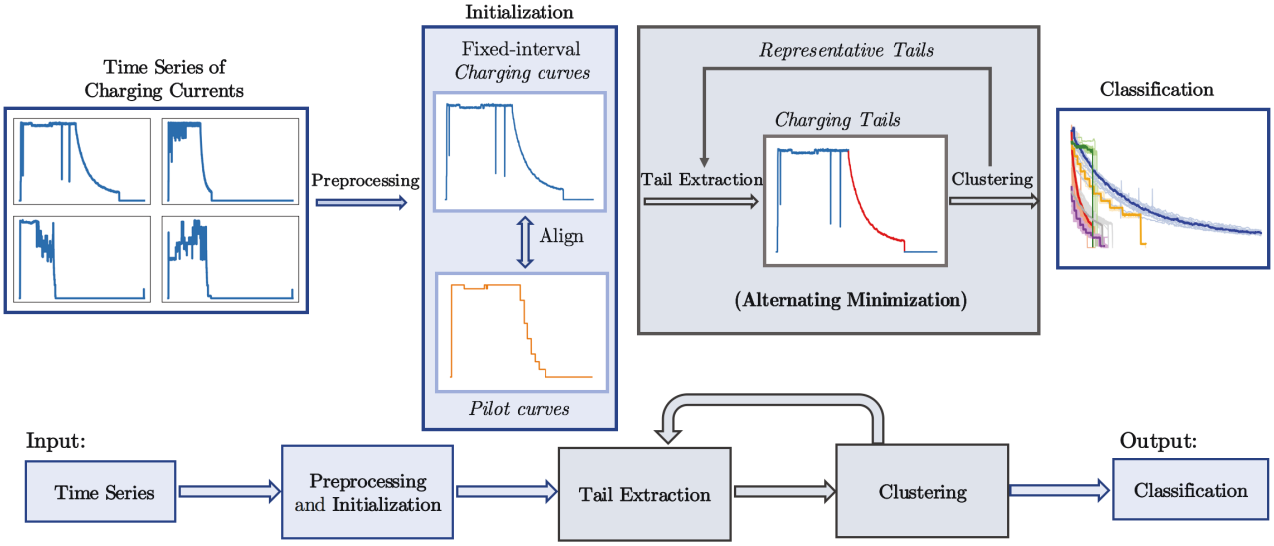

Fig. 3. The classification method introduced in this paper.

3) Extraction by matching: Our third method assumes that all charging tails from the same EV have similar properties such as duration and shape. Before the iterative steps, suppose that for a fixed user, we are able to obtain an initial charging tail $\mathbf{x}^{(1)}$, *e.g.*, using the two methods above. This $\mathbf{x}^{(1)}$ is used as a "template" to extract the tails of all other charging curves of the same user. Then, we go through the subsequences of the charging curve that have the same length as the template, and find a charging tail with improved noise robustness. Suppose we obtain a tail representative $\mathbf{x}$ for a fixed user. For the remaining sessions i of the same user, we minimize the Euclidean distance $d_{\mathrm{ED}}\left(\mathbf{x}, \mathbf{x}_i^{(1)}\right)$ over all consecutive subsequences $\mathbf{x}_i^{(1)}$ of the charging curve $\mathbf{s}_i$ that have the same length as $\mathbf{x}$. In this way, we use the three extraction rules jointly to compute the initial tails $\mathbf{x}_1^{(1)}, \ldots, \mathbf{x}_n^{(1)}$ in Algorithm 1. Fig. 4 illustrates the idea and effectiveness of this approach.

Besides speeding up the initialization, the third approach is also used as the TE step as an approximation of the optimization in (2). At the ℓ-th iteration, by setting the medoid (tail representative) $\mathbf{c}_k^{(\ell)}$ of the k-th cluster that the charging curve $\mathbf{x}_i$ is classified into as the template[1] and using the Euclidean distance as the distance function, we approximate the optimization in (2) for the ℓ-th iteration:

$$\widehat{\mathbf{x}}_i^{(\ell+1)} = \arg\min_{\mathbf{x}} d_{\mathrm{ED}}\left(\mathbf{c}_k^{(\ell)}, \mathbf{x}\right) \tag{4}$$

where the minimization is over all $\mathbf{x} \in \mathcal{X}_i\left(\mathbf{c}_k^{\ell}\right)$ and $\mathcal{X}_i\left(\mathbf{c}_k^{\ell}\right)$ is the set containing all consecutive subsequences of the charging curve $\mathbf{s}_i$ that have the same length as $\mathbf{c}_k^{(\ell)}$.

[1]In our experiments (elaborated in Section IV), for improving efficiency, we implement a simplified TE, wherein we focus on the medoid of the cluster that the charging curve $\mathbf{s}_i$ for session i belongs to and remove the minimization over k in (2). This modification does not affect the local convergence property stated in Theorem II.1.

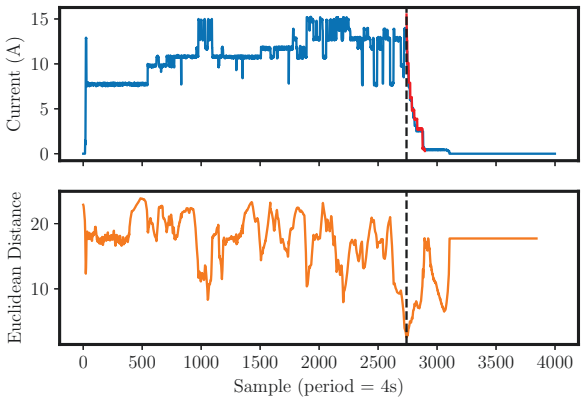

Fig. 4. An example of *extraction by matching*. The red subsequence $\mathbf{x}_1$ is a template with *userID* 409, which is extracted from the first session $\mathbf{s}_1$ of this user. The figure below visualizes the change of Euclidean distance of the second session $\mathbf{s}_2$ with respect to $\mathbf{x}_1$. The black vertical line indicates the best matching location in $\mathbf{s}_2$ for $\mathbf{x}_1$ and the tail $\mathbf{x}_2$ can be found correspondingly despite the slight difference of both tails.

C. Tail clustering

Time series clustering is a well-studied problem; see [7] for a review and [8] for a detailed experimental comparison. One of the main problems considered in the literature is determining the distance/similarity between time series. Based on their own applications, a variety of similarity distance metrics have been proposed, including the Euclidean distance [9] for stock price movements clustering, the edit distance [10] for trajectory clustering and the cross correlation [11] for electrocardiogram time series clustering, *etc*. However, most of the existing metrics require that the two sequences have the same length. As an exception, dynamic time warping (DTW) [12] is able to calculate the distance between two sequences with different lengths. However, it is computationally

expensive. For clustering tails, we introduce a penalty term to the Euclidean distance and our experiments show that the new distance function (defined as the MED in (5)) surpasses the others for charging time series clustering. We use similarity based clustering techniques for solving the minimization in (3). Tails of varying lengths are clustered in two steps: (a) similarity matrix construction (b) similarity based clustering.

1) Similarity matrix construction: The lengths of tails extracted from the charging curves of different EVs are generally different. This creates difficulty in comparing two tails as the standard Euclidean distance is defined for two vectors of the same length. We compare three different distance definitions for tails of different lengths and more results can be found in Section IV. The first method simply pads the shorter tail with zeros to make two tails the same length so their distance is the standard Euclidean distance (ED). The second method uses a distance function defined as follows. Suppose $\mathbf{x} \in \mathbb{R}^s$ and $\mathbf{y} \in \mathbb{R}^l$ with $s \leq l$. Their corresponding *modified Euclidean distance* (MED) is

$$d_{\text{MED}}(\mathbf{x}, \mathbf{y}) := \min \left\{ d_{\text{ED}}(\mathbf{x}, \mathbf{y}(\leq s)), \right. \\ \left. d_{\text{ED}}(\mathbf{x}, \mathbf{y}(\geq l - s + 1)) \right\} + \lambda |l - s| \quad (5)$$

where $d_{\text{ED}}(\cdot, \cdot)$ is the Euclidean distance and $\mathbf{y}(\leq s)$ and $\mathbf{y}(\geq l - s + 1)$ represent the first s and last s coordinates of $\mathbf{y}$, respectively. The *penalty parameter* $\lambda > 0$ can be tuned. By default we set it to 1. Note that the distance function MED in (5) may not satisfy the triangle inequality. The third method uses the dynamic time warping (DTW) defined in [12], [13]. The clustering results obtained via the ED with zero padding technique, the MED defined above and the DTW are compared in Fig. 5, with more details in Section IV-A.

2) Similarity based clustering: For similarity based clustering, we apply the spectral clustering [14]–[16] as the heuristic for approximating the minimization in (3).

IV. Clustering, Applications and Discussions

A. Clustering evaluation

In this section, we evaluate the proposed method (shown in Fig. 3) on ACN-Data [6]. We use the dataset from JPL from Sep. 2018 to Dec. 2018 as the training data, which contains 2933 claimed sessions from 195 users.[2] In preprocessing (see Section III-A), we resample the data at a time resolution of $\delta = 4$ seconds.

We use two evaluation metrics to find the number of clusters K. The first is the *silhouette coefficient* [17], which takes a value in $[-1, 1]$. A higher silhouette coefficient indicates better clustering performance. The second is the *correctly classified percentage*. Recall that each tail is associated with a *userID* (see Table I). We evaluate the clustering quality by checking if the tails with the same *userID* are consistently grouped into the same cluster.[3] A tail is considered *correctly classified* if it is clustered into a group wherein the majority of the tails have the same *userID* as the considered tail.

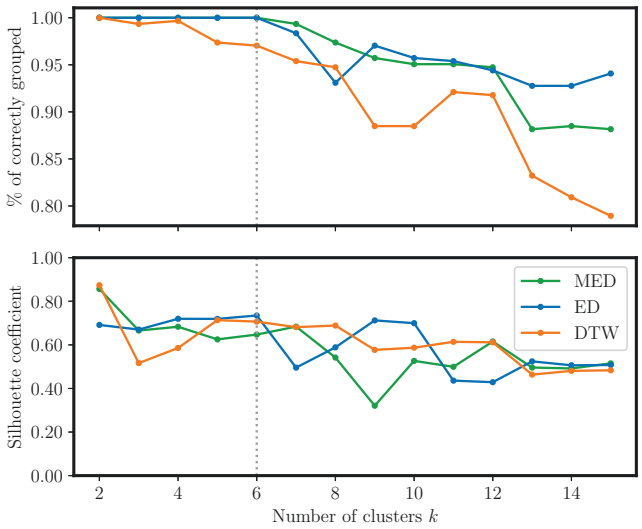

Fig. 5. The performance for different number of clusters using three different distance functions – Euclidean distance (ED), Modified Euclidean distance (MED), Dynamic Time warping distance (DTW).

The evaluation results for three different distance functions – the modified Euclidean distance (MED), Euclidean distance (ED), and dynamic time wrapping (DTW) are shown in Fig. 5. We use $\lambda = 1$ for MED. For distance to similarity conversion, we use the Gaussian kernel $\kappa(d) = \exp\left(-d^2/\theta\right) \in (0, 1]$ where d is the pairwise distance between two sequences and $\theta > 0$ is a tuning parameter. In particular, we choose $\theta = 20$ for MED and ED and $\theta = 110$ for DTW. It can be seen that $K = 6$ is a good choice of number of clusters for this dataset, as it is the largest value at which all the tails are correctly classified for both ED and MED. In addition, the silhouette coefficient is relatively high for $K = 6$. Fixing $K = 6$, the clustering results for different distance functions are visualized in Fig. 6. It can be seen that using MED, the six clusters are well-separated and the corresponding medoids provide informative patterns for charging tails. In Fig. 7, we project the tails to a two-dimensional space using the t-distributed stochastic neighbor embedding (t-SNE) and MED. It demonstrates the hierarchical relationship between the groups of users and the clusters for our training data.

B. Charging behavior prediction

The ability to classify charging behavior can enable both offline and online applications in the future (see Section V). One of the building blocks for these applications will be the use of cluster representatives for prediction. In this subsection, we illustrate its accuracy.

[2]More than a half of the users have less than 12 charging sessions during the period. In the clustering experiment, we only consider the 35 users with more than 30 sessions. Out of the 35 users, 16 of them have sufficient number of charging curves with tail-like features. Our experiments used the 304 charging curves from these 16 users.

[3]It is possible that the same *userID* exhibits different charging patterns. This may occur if the user changes her EV or owns more than one EV. But as shown in Table III, such scenario is rare.

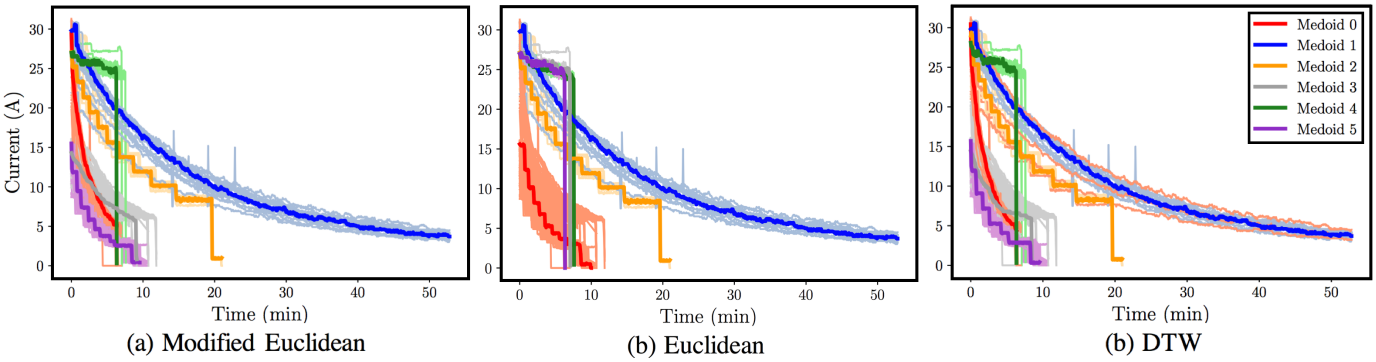

Fig. 6. Visualization of $K = 6$ clusters for MED, ED and DTW. Tails are within the same cluster if they have the same color and the tail representatives (medoids) are emphasized.

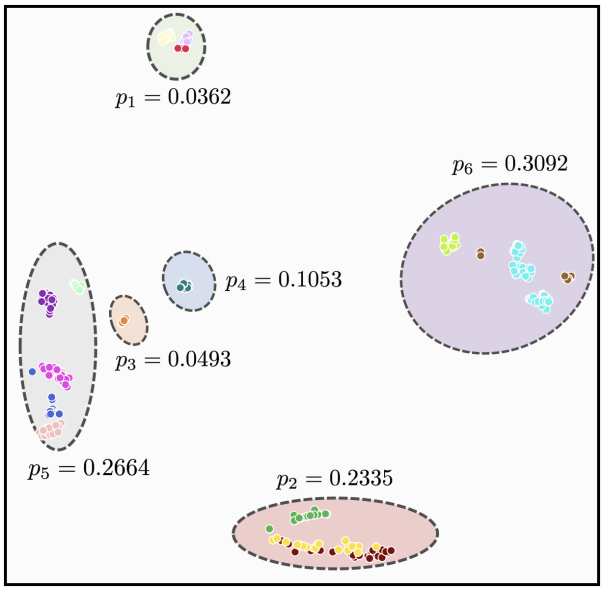

Fig. 7. Two-dimensional visualization of our clustering results with $K = 6$ clusters. Tails for different users are colored differently. The clusters' colors are consistent with those used in Fig. 6. The marginal probabilities $p_1, \ldots, p_6$ represent the portions of charging sessions falling into the six clusters.

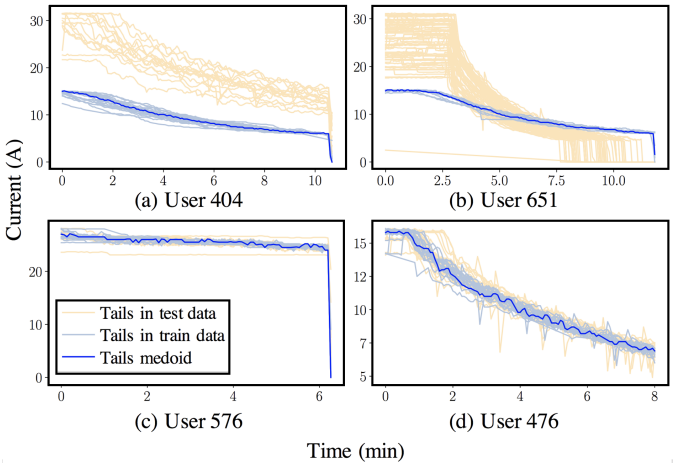

Fig. 8. Examples of the training and testing data (tails) for four users. Sub-figures (a) and (b) are the tails of the two users with poor prediction performance (highlighted in blue in Table II). The poor prediction performance is due to the fact that the tails in the training data are very different from those in the testing data. Sub-figures (c) and (d) are examples where the tail representatives achieve high-quality prediction performance. Tails in the training data and those in the testing data are similar.

The training data is the same as in Section IV-A and the testing data contains 731 tails for 1441 sessions collected from Jan. 2019 to Aug. 2019. We use the tail representatives of the training data obtained using our framework in Fig. 3 to predict the behavior of the charging tails of the testing data. Denote by $\mathbf{s}$ a real charging curve in the testing data and $\widehat{\mathbf{x}}$ the estimated tail. We consider two situations – with and without the knowledge of *userID*, and the results are shown in Table II and Table III respectively. We evaluate the prediction quality using the following three metrics. The first metric is the *coefficient of determination* (R^2) (generalized in our case for comparing two sequences of different lengths) defined as:

$$R^2_{\text{Predict}}(\mathbf{s}, \widehat{\mathbf{x}}) := \min_{\mathbf{x}} \left\{ 1 - \frac{\sum_{t=1}^{r}(x_t - \widehat{x}_t)^2}{\sum_{t=1}^{r}(x_t - \bar{x})^2} \right\} \qquad (6)$$

where the minimization is over all consecutive subsequences $\mathbf{x}$ of the charging curve $\mathbf{s}$ that have the same length as $\widehat{\mathbf{x}}$ and $\bar{x} = \sum_{t=1}^{r} x_t/n$ and r is the length of $\mathbf{x}$ and $\widehat{\mathbf{x}}$. It ranges from $(-\infty, 1]$ and the larger the better. A negative value indicates that performance is worse than the arithmetic mean mean. Our second metric is the *root mean square error (RMSE)* that is useful for measuring scale-dependent prediction error. The last metric is the *mean absolute error (MAE)*. Similar to (6), the last two metrics are also generalized with an additional minimization over consecutive subsequences of charging curves in the testing data.

Table II shows the *userID*-based prediction results. Each tail representative (medoid) corresponds to each group of users. As can be observed from the results, except for user 404 and user 651, the tail representatives of the other 14 users can well predict the charging tail behavior in incoming sessions for the same user. Fig. 8 visualizes the training tails, testing tails and tail representatives of 4 users, including the two users with high prediction error. Note that the charging tails of user 404 exhibit two distinct groups, one is from Sep. 2018 to Dec.

TABLE II
PREDICTION RESULTS WITH USER TAIL REPRESENTATIVE

userID	R^2	RMSE	MAE
322	0.6464± 1.2509	0.9306 ± 1.1596	0.7947 ± 1.1773
334	0.8783± 0.4765	1.3840 ± 1.3656	0.9954 ± 1.2012
368	0.6812 ± 0.6259	2.4878 ± 1.9991	2.0742 ± 1.7777
371	0.8615 ± 0.2040	0.8735 ± 0.6297	0.6946 ± 0.5655
374	0.9329 ± 0.0571	1.3872 ± 0.5800	0.9128 ± 0.4418
404	-11.906 ± 4.9304	10.522 ± 2.1957	10.230 ± 2.1357
405	0.8335 ± 0.1253	1.2936 ± 0.4793	0.9011 ± 0.3733
406	0.8917 ± 0.1315	0.5243 ± 0.2889	0.4082 ± 0.2305
409	0.9078 ± 0.0464	0.9412 ± 0.2310	0.6423 ± 0.1853
476	0.9509 ± 0.0757	0.5369 ± 0.3062	0.4077 ± 0.2536
551	0.9199 ± 0.1256	1.4938 ± 1.0355	1.1755 ± 0.7750
576	0.9209 ± 0.0249	0.6258 ± 0.1136	0.4802 ± 0.1178
577	0.9150 ± 0.0749	1.0301 ± 0.4422	0.6683 ± 0.2736
592	0.8699 ± 0.2607	0.7686 ± 0.6002	0.5394 ± 0.4607
607	0.9506 ± 0.0936	1.0141 ± 0.7940	0.8195 ± 0.6497
651	-3.5447 ± 1.9932	6.7702 ± 1.5415	5.4010 ± 1.0258

TABLE III
PREDICTION RMSE WITH CLUSTER REPRESENTATIVE

UserID	MED	ED	DTW
322	2.7170 ± 0.8737	**0.9306 ± 1.1596**	2.7363 ± 0.9032
334	**1.1880 ±1.4109**	4.3331 ± 1.0066	2.8291 ± 1.7102
368	**2.6745 ± 2.0675**	2.7885 ± 1.2932	2.8450 ± 1.5884
371	**0.8266 ± 0.5495**	3.1160 ± 0.7986	**0.8266 ± 0.5495**
374	**1.3872 ± 0.5800**	**1.3872 ± 0.5800**	3.8939 ± 2.4183
404	**9.4865 ± 2.0709**	13.4698 ± 2.1212	**9.4865 ± 2.0709**
405	**1.2805 ± 0.5074**	1.4343 ± 0.5525	1.3289 ± 0.4918
406	**1.4506 ± 0.4911**	3.0573 ± 0.1223	**1.4506 ± 0.4911**
409	1.0244 ± 0.1878	1.6821 ± 0.5381	**0.9960 ± 0.1940**
476	**1.5438 ± 0.3304**	4.2103 ± 0.3307	**1.5438 ± 0.3304**
551	**1.5861 ± 1.0745**	**1.5861 ± 1.0745**	4.8002 ± 3.3426
576	2.5593 ± 0.0552	**0.7033 ± 0.1582**	2.6073 ± 0.0357
577	0.8972 ± 0.2218	1.4100 ± 0.5203	**0.8832 ± 0.2140**
592	**0.7682 ± 0.5871**	0.7686 ± 0.6002	0.8690 ± 0.5895
607	**1.0393 ± 0.6984**	4.2828 ± 0.9654	2.7691 ± 1.4346
651	**4.7786 ± 0.5789**	5.4719 ± 1.8010	**4.7786 ± 0.5789**

2018 (tails colored in light blue) and the other is from Jan. 2019 onward (tails colored in light orange); similarly for user 476. Unlike for the other users, the tails in the training data are very different from those in the testing data for user 404 and 651. Ignoring the user labels, Table III compares the prediction RMSE using the most similar cluster representative from the 6 clusters obtained for three different distance functions – MED, ED and DTW. In this case, the estimate is the tail representative of the cluster to which the charging curve in the testing data belongs. The best distance function for each user is highlighted in bold. MED is the best for most of the cases. In addition, Tables II and III show that the cluster representatives with MED achieves comparable and even better prediction than user representatives, indicating the existence of charging tail patterns.

C. Charging stage decision

In the remainder of our experimental results, we consider a real-time binary decision problem on whether an EV is in the absorption stage (AS) (see Section II-C for more details of the AS) or not. Our training data remains the same. In particular, for the testing data, we choose the user with ID 476 as an example, and manually label the start time t_s and end time t_e of the AS for each of the charging sessions since Jan. 2019. There are $n = 38$ out of 46 sessions in total that contain tails. The MAE is used for deciding the charging stage. Let ε_{MAE} be the *error threshold*. At time $t \in \mathcal{T}$, denote by m the number of samples that can be used in our decision. Equivalently, m is the time delay that are allowed for deciding if at time t the EV enters the AS. The decision rule in our experiments is that if $d_{\text{ED}}\left(\mathbf{s}\left(t:t+m\right), \mathbf{c}\left(\leq m\right)\right) \leq \varepsilon_{\text{MAE}}$, then we claim that the EV is in the AS; otherwise the EV is not in the AS where $\mathbf{s}\left(t:t+m\right) := s(t), \ldots, s(t+m)$ and $\mathbf{c}\left(\leq m\right) := c(1), \ldots, c(m)$. We set $\varepsilon_{\text{MAE}} = 0.7$ in the tests.

Fig. 9 shows the trade-offs between the decision accuracy and the number of samples. In particular, in Fig. 9, the *average accuracy* is defined as

$$\sum_{i=1}^{n} \frac{\text{TP}_m(\mathbf{s}_i) + \text{TN}_m(\mathbf{s}_i)}{\text{TP}_m(\mathbf{s}_i) + \text{TN}_m(\mathbf{s}_i) + \text{FN}_m(\mathbf{s}_i) + \text{FP}_m(\mathbf{s}_i)}$$

where $\text{TP}_m(\mathbf{s}_i)$, $\text{FP}_m(\mathbf{s}_i)$, $\text{TN}_m(\mathbf{s}_i)$ and $\text{FN}_m(\mathbf{s}_i)$ are the numbers of true positive, false positive, true negative and false negative decisions for the charging stage decision of a charging curve $\mathbf{s}_i$ with m samples. The *average sensitivity* and *average precision* are defined similarly as

$$\sum_{i=1}^{n} \frac{\text{TP}_m(\mathbf{s}_i)}{\text{TP}_m(\mathbf{s}_i) + \text{FN}_m(\mathbf{s}_i)} \text{ and } \sum_{i=1}^{n} \frac{\text{TP}_m(\mathbf{s}_i)}{\text{TP}_m(\mathbf{s}_i) + \text{FP}_m(\mathbf{s}_i)},$$

respectively. Both the average precision and the average sensitivity grow with the number of samples m.

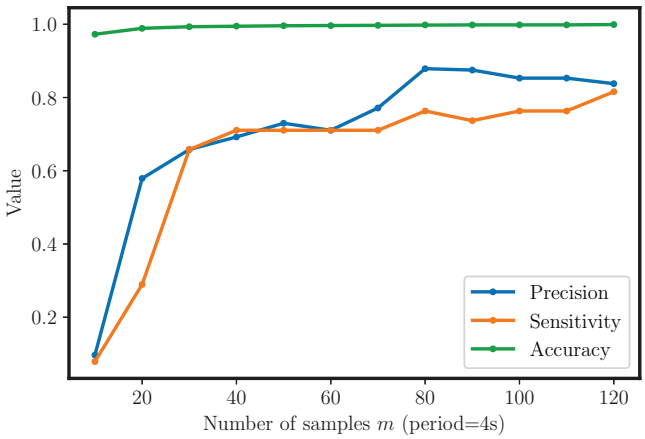

Fig. 9. Trade-offs between the number of samples m and the accuracy, sensitivity and precision.

V. CONCLUSION

We have presented an iterative clustering method to classify EV charging time series and illustrated its performance using ACN-Data, a fine-grained charging dataset recently made publicly available. Our analysis shows that, even though the number of charging curves is large, they can be accurately

classified into a small number of types. Moreover, the cluster representatives can be used for effective prediction.

This opens up potentials for future applications. For instance, a natural statistical EV model consists of $(\mathbf{c}_k, p_k, k = 1, \ldots, K)$ where $\mathbf{c}_k$ is the tail representative and p_k is the marginal probability that an EV arrival is of type $k = 1, \ldots, K$ as exemplified in Fig. 7. This model can be useful for planning purposes and for simulations, *e.g.*, to determine the capacity of electric infrastructure supplying a large-scale EV charging facility. For another instance, online optimization of EV charging can be implemented as a model predictive control (MPC) where a forward optimization problem is solved in each control interval (*c.f.*, [4], [5]). The representative for each individual user can be used as prediction to improve the performance of MPC. Moreover, the ability to decide the charging stage in real time as illustrated in Section IV-C can be helpful to online scheduling. Yet another application is to use the representative tail $\mathbf{c}_k$ to detect abnormal battery behavior in real-time and alert the drivers, or charging facilities, or EV manufacturers.

This paper presents the first analysis of the fine-grained charging data in ACN-Data and develops a systematic method to learn battery behavior from the data. It has several limitations that motivate extensions. First, our current method works well only with charging curves that exhibit relatively clean tail behavior. Additional techniques are needed to extract useful information from other charging curves. Second, our current method is offline. It would be useful to extend it to an online setting, for continuous improvement of classification performance and adaptation to changing EV behavior. Such an online method will be useful as the building block for many online applications. Here theories and algorithms in statistical detection and signal processing will prove to be helpful. Third, we model battery behavior by the representative tail $\mathbf{c}_k$ as functions of time. More detailed battery models can be developed using $\mathbf{c}_k$ and other information such as the energy capacities of the batteries and the voltage time series, *e.g.*, their current and voltage behavior in the absorption stage as functions of their states of charge. Finally, it would be interesting to develop a tractable mathematical model of the classification framework shown in Figure 3 and formally prove its convergence and optimality properties.

REFERENCES

[1] IEA 2019, "Global EV outlook 2019," available at www.iea.org/publications/reports/globalevoutlook2019/.

[2] J. Van Mierlo, P. Van den Bossche, and G. Maggetto, "Models of energy sources for ev and hev: fuel cells, batteries, ultracapacitors, flywheels and engine-generators," *Journal of power sources*, vol. 128, no. 1, pp. 76–89, 2004.

[3] A. Seaman, T.-S. Dao, and J. McPhee, "A survey of mathematics-based equivalent-circuit and electrochemical battery models for hybrid and electric vehicle simulation," *Journal of Power Sources*, vol. 256, pp. 410–423, 2014.

[4] G. Lee, T. Lee, Z. Low, S. H. Low, and C. Ortega, "Adaptive charging network for electric vehicles," in *GlobalSIP*, 2016.

[5] Z. J. Lee, D. Chang, C. Jin, G. S. Lee, R. Lee, T. Lee, and S. H. Low, "Large-scale adaptive electric vehicle charging," in *SmartGridComm*. IEEE, 2018.

[6] Z. J. Lee, T. Li, and S. H. Low, "ACN-Data: Analysis and Applications of an Open EV Charging Dataset," in *Proceedings of the Tenth International Conference on Future Energy Systems*, ser. e-Energy19, Jun. 2019.

[7] S. Aghabozorgi, A. S. Shirkhorshidi, and T. Y. Wah, "Time-series clustering–a decade review," *Information Systems*, vol. 53, pp. 16–38, 2015.

[8] X. Wang, A. Mueen, H. Ding, G. Trajcevski, P. Scheuermann, and E. Keogh, "Experimental comparison of representation methods and distance measures for time series data," *Data Mining and Knowledge Discovery*, vol. 26, no. 2, pp. 275–309, 2013.

[9] C. Faloutsos, M. Ranganathan, and Y. Manolopoulos, *Fast subsequence matching in time-series databases*. ACM, 1994, vol. 23, no. 2.

[10] J.-G. Lee, J. Han, and K.-Y. Whang, "Trajectory clustering: a partition-and-group framework," in *Proceedings of the 2007 ACM SIGMOD international conference on Management of data*. ACM, 2007, pp. 593–604.

[11] J. Paparrizos and L. Gravano, "k-shape: Efficient and accurate clustering of time series," in *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*. ACM, 2015, pp. 1855–1870.

[12] E. Keogh and C. A. Ratanamahatana, "Exact indexing of dynamic time warping," *Knowledge and information systems*, vol. 7, no. 3, pp. 358–386, 2005.

[13] D. J. Berndt and J. Clifford, "Using dynamic time warping to find patterns in time series." in *KDD workshop*, vol. 10, no. 16. Seattle, WA, 1994, pp. 359–370.

[14] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in neural information processing systems*, 2002, pp. 849–856.

[15] J. Shi and J. Malik, "Normalized cuts and image segmentation," *Departmental Papers (CIS)*, p. 107, 2000.

[16] C. Sun, T. Li, and V. O. Li, "Robust and consistent clustering recovery via sdp approaches," in *2018 IEEE Data Science Workshop (DSW)*. IEEE, 2018, pp. 46–50.

[17] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.