Applied Energy 163 (2016) 134–141

Contents lists available at ScienceDirect

Applied Energy

journal homepage: www.elsevier.com/locate/apenergy

Forecasting the EV charging load based on customer profile or station measurement?

Mostafa Majidpour [a,*], Charlie Qiu [a], Peter Chu [a], Hemanshu R. Pota [b], Rajit Gadh [a]

[a] Smart Grid Energy Research Center, UCLA, Los Angeles, CA, USA
[b] School of Engineering & Information Technology, The University of NSW, Canberra, ACT 2610, Australia

HIGHLIGHTS

- We compare the forecasting of the EV charging load based on two different datasets.
- Customer profile dependent dataset is prone to privacy invasion.
- Station measurement dataset is directly measured from charging outlets.
- Results show customer profile based prediction is faster due to less preprocessing.
- We found that both datasets yield comparable forecasting error.

ARTICLE INFO

Article history:
Received 18 April 2015
Received in revised form 26 October 2015
Accepted 31 October 2015
Available online 19 November 2015

Keywords:
Electric Vehicle
Privacy
Time series
Load forecasting

ABSTRACT

In this paper, forecasting of the Electric Vehicle (EV) charging load has been based on two different datasets: data from the customer profile (referred to as charging record) and data from outlet measurements (referred to as station record). Four different prediction algorithms namely Time Weighted Dot Product based Nearest Neighbor (TWDP-NN), Modified Pattern Sequence Forecasting (MPSF), Support Vector Regression (SVR), and Random Forest (RF) are applied to both datasets. The corresponding speed, accuracy, and privacy concerns are compared between the use of the charging records and station records. Real world data compiled at the outlet level from the UCLA campus parking lots are used. The results show that charging records provide relatively faster prediction while putting customer privacy in jeopardy. Station records provide relatively slower prediction while respecting the customer privacy. In general, we found that both datasets generate comparable prediction error.

1. Introduction

The most distinct feature of the smart grid is its extensive use of information and communication technologies to improve the efficiency and reliability of the generation and distribution of electricity. A large volume of information is gathered from different meters that might be sufficient to reveal the behavior of different players such as suppliers and consumers. This calls for a privacy concern as is pointed out [1].

Electric Vehicle (EV) charging related data is no exception to privacy issues and has its own problems. One such problem is the large battery size in today's EVs, which may require a relatively large amount of charging time depending on the charging station capabilities. The long charging time may obligate EV owners to charge their EVs in places other than their household, including public charging stations or charging stations at their work place. This implies that not only utilities have access to charging data through household chargers, but also charging station administrators in work places and public stations have access to them. These data, when used for different analysis in utility or public charging station operation or planning, might expose information such as the pattern of entrance and exit times of the customers from charging lots or their home, hence risking their privacy. To exemplify the battery size, consider Chevrolet Volt, Nissan Leaf, and Tesla, with a battery size of 16.5 kW h, 24 kW h, and up to 85 kW h, respectively [2]. With Level 1 household chargers (16A at 230VAC which delivers 3.3 kW) it will take the Nissan Leaf around 8 h, and Tesla (Model S85) around 25 h to charge completely.

In this paper, our application is forecasting (predicting) EV charging load based on historical charging data. We have two available datasets: charging record that comes from anonymous

* Corresponding author.
E-mail address: mostafam@ucla.edu (M. Majidpour).

http://dx.doi.org/10.1016/j.apenergy.2015.10.184
0306-2619/

customer profiles and station records that come from measurements (voltage, current, etc.). Either one of them can be used for building a load time series and hence forecasting at the outlet level.

One might wonder since the charging records are anonymized, there is no threat to customer's privacy. However, anonymizing might not be enough, as in a famous incident, the medical record for the then governor of Massachusetts was extracted easily from anonymous medical records when combined with voter registration rolls [3]. The medical records were anonymous but they had sex, ZIP code, and birth-date of patients. This incident shows that even anonymity is not enough and anonymous data might still be revealing when combined with other datasets.

Therefore, we deal with two datasets: The charging record comes from customer profiles, and, as pointed out earlier, it is prone to privacy issues. On the other hand, the station record does not have any information about specific customers and hence protects customer privacy. We compare the accuracy and speed of the prediction process using these two types of records. Specifically, for EV charging data, we investigate the potential increase in prediction accuracy and speed, as a tradeoff of endangering customer privacy. Interestingly, we found that prediction accuracy is not significantly increased while using the privacy-jeopardizing dataset (charging records). To our knowledge, this type of comparison has not been done in this context.

The rest of this paper is organized as follows: Section 2 provides a brief review of the existing literature, Section 3 formulates the problem, Section 4 reviews the prediction algorithms applied on time series based on both of the datasets. Section 5 discusses the structure of each dataset from the charging stations at the University of California, Los Angeles (UCLA) parking lots and the preprocessing stages to convert each of them into a time series. Section 6 reports the result of applying the prediction algorithms on each of the time series and then analyses the results with nonparametric statistical tests to investigate statistically meaningful differences. Section 7 provides the conclusion and future work.

2. Literature review

Privacy has been an important issue in smart grids and it is one of the factors holding people back from participating in the use of these new technologies [4].

There are various levels of invasion of privacy in smart grid context. For example, at the smart household level, the different ways that household privacy might be invaded are: access to power consumptions records, presence of different players with potential access to data such as service provider and distribution operator in the economic smart grid, using wireless communication technology between devices that might make communications vulnerable, accessing energy devices from the internet, and third parties that are not involved in any part of power generation and distribution but monitor the customer usage with customers' approval [5].

According to [6], the current resolution of smart meter data (usually between 15 min and 1 h) invades customer privacy and the data might not be necessary for most of the smart grid planning and distribution functions. Some other research still relies on anonymous data to protect the privacy where the pattern of the EV customer driving times is used for designing an optimal charging algorithm [7].

There have been various suggestions on how to preserve privacy. Some of them are based on the idea of aggregating the data instead of using individual data. For example, in [8], building energy usage is investigated and only the aggregated data, instead of individual data, is used for analysis. Similarly, instead of individual EV charging data, the aggregated data of EV loads has been used for coordinating the EV charging operation in [9].

On the other hand, centralized and distributed algorithms for the routing of the information flows which preserve the privacy based on cryptographic methods is proposed in [10]. Cryptographic methods are also used in [11] to perform privacy preserving bill calculations. All these methods fall under Secure Signal Processing (SSP) methods which protect the sensitive data by encryption and provide tools to analyze the data under the applied encryption [12].

According to [13], privacy is exposed even when relatively infrequent measurements are acquired, and on the other hand, the energy management system with battery can protect the customer privacy. The role of the battery and its use in more effectively protecting privacy is also discussed in [14].

Lastly, the above articles focus mostly on protecting privacy at the measurement level. Another way to protect the privacy is at the data mining algorithm level. Privacy preserving machine learning algorithms started to become more important in early 2000s [15,16]. In subsequent years, privacy preserving versions of different machine learning algorithms such as nearest neighbor [17], Bayes classifiers [18], Support Vector Machines [19], and logistic regression [20] were introduced in literature to address privacy-preserving in algorithmic level.

There is a rich literature for time series forecasting in various disciplines. Ref. [21] provides a comprehensive review of different models. Machine Learning algorithms have also been successfully employed in the forecasting realm [22]. In the current work, we compare four machine learning based prediction algorithms on two time series built from different measurements of one phenomenon, one of which contains privacy-insensitive information whereas the other does not. We show that the dataset without privacy-sensitive information allows us to make equally accurate charging load prediction compared to the other dataset, thus precluding the need for privacy-preserving data mining techniques.

3. Problem formulation

The objective is to predict the energy consumption in the next 24 h at each charging outlet based on two datasets namely charging record and station record, and comparing the two approaches. Formally, we assume there is a function relating the predicted available energy and the past consumed energy:

$$\widehat{E}(t) = f(E(t-1), E(t-2), \ldots) \tag{1}$$

where $E(t)$ is the actual energy consumption at time t, $\widehat{E}(t)$ is the prediction of the energy consumption at time t, and $(E(t-i))$ indicates the past energy consumption records.

As is the usual practice in forecasting, we are interested in finding an estimation of $E(t)$ according to a particular performance (or error) criterion. To this end, we have chosen Symmetric Mean Absolute Percentage Error (SMAPE). For day i, the SMAPE is defined as:

$$SMAPE(i) = \frac{1}{H} \sum_{t \in dayi} \frac{|E(t) - \widehat{E}(t)|}{E(t) + \widehat{E}(t)} \times 100, \tag{2}$$

where H is the horizon of prediction in a given day ($H = 24$ in this paper).

In this paper the most recent ten percent of the data is used to evaluate the performance of the algorithm (test dataset). Note that the test dataset is not used in either the parameter selection or training phase.

We use the notation $\hat{\boldsymbol{E}}(t)$ as the vector of prediction for the next 24 h ending at t (Fig. 1a). Also, let $t_r = \{1, 2, \ldots, N_{tr}\}$ and $t_s = \{N_{tr}+1, \ldots, N\}$ be two sets of indices for the training and test sets, respectively. Later on, in the parameter selection phase, parts

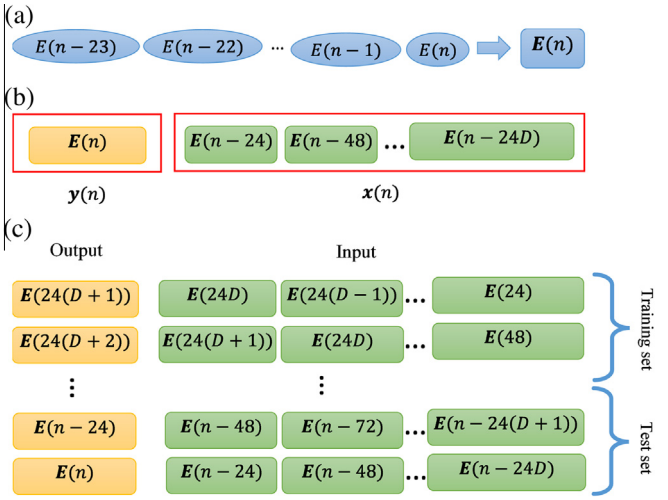

Fig. 1. (a) Energy consumption vector ($\boldsymbol{E}$) for 24 h, (b) input–output pairs and division of data into training and test sets, and (c) labeling inputs as x and outputs as y.

of the training set will be treated as the validation set. The different methods used to select the validation set are further explained in the parameter selection section.

4. Applied algorithms

Four prediction algorithms have been briefly described here [23]. These algorithms are some of commonly used machine learning algorithms in different disciplines [24]. They are selected based on [25] to compare the prediction process between two types of datasets (charging record and station record).

4.1. K-Nearest Neighbor

This algorithm is a variation of a well-known algorithm in the machine learning community named K-Nearest Neighbor (kNN) [26]. Based on the kNN algorithm, each sample (training, test or validation) is composed of input and output pairs. In our application, the output is the energy consumption for the next 24 h, $\boldsymbol{y}(t) = \boldsymbol{E}(t)$, and the input is the concatenation of the consumption records for up to D prior days, $\boldsymbol{x}(t) = \{\boldsymbol{E}(t-24), \boldsymbol{E}(t-48), \ldots, \boldsymbol{E}(t-24D)\}$ (Fig. 1c). This concatenation repeats for all days: if there are N days in the dataset, there will be $N - D + 1$ of these input–output pairs (Fig. 1b). The total number of data points is$n = 24N$. Now, in order to find an estimate for $\boldsymbol{y}(t_{s^*})$ where $t_{s^*} \in t_s$ is an instance of test set indices, first, the dissimilarity between $\boldsymbol{x}(t_{s^*})$ and all other $\boldsymbol{x}(t_r)$ that belong to the training set is computed. There are various dissimilarity measures and Euclidian distance is popular; however, it has been shown in [25] that dissimilarities based on the dot product have less prediction error than Euclidian distance. As a result, we use the time weighted dot product (TWDP) based dissimilarity in this paper. After determining the k closest $\boldsymbol{x}(t_r)$ to $\boldsymbol{x}(t_{s^*})$, the average of their corresponding $\boldsymbol{y}(t_r)$ is generated as $\boldsymbol{y}(t_{s^*})$. Based on [25], the k has been selected to be equal to one. Fig. 2 illustrates the TWDP based Nearest Neighbor algorithm where $\boldsymbol{dis}[j]$ refers to dissimilarity between $\boldsymbol{x}(t_{s^*})$ and $\boldsymbol{x}(j)$, and $\boldsymbol{TW}$ is the linear time weighting vector:

$$\boldsymbol{TW} = [1, 1+\Delta, 1+2\Delta, \ldots, D]$$

with $\Delta = (D-1)/(24D-1)$ where D is the depth of input which will be determined through cross validation.

4.2. Modified Pattern Sequence-based Forecasting (MPSF)

Pattern Sequence-based Forecasting (PSF) is a successful energy price forecasting algorithm that was first introduced in [27] and was later improved in [28]. The idea is based on assigning each 24 h set, i.e. a day, to a cluster and then the forecast is based on the cluster labels rather than actual values in each day. By clustering, the dimension of each day reduces to one (label of the cluster in which the day belongs) instead of 24 (values for 24 h). It also adds robustness by substituting real values (e.g., power consumption) with integer numbers (cluster labels). The modified version of the PSF (MPSF) was introduced in [29] and has shown improved results over PSF.

The first step in applying the MPSF algorithm is to find the clustering method and the optimum number of clusters. In this study, we use the k-means clustering algorithm similar to the one in [27]. In order to determine the optimum number of clusters, k-means clustering is performed on the training dataset for a k ranging from about 10–100% of the number of unique instances in the training dataset. The k with the highest validity index is selected. The validity index is a clustering statistic that helps to find the optimum number of clusters. The silhouette index, SI, is the validity index used in [27] and is explained in detail in [29]. SI can range between −1 and +1 where a larger value indicates more suitable clustering. The number of clusters that maximizes the SI is selected as optimum.

After clustering, a 24 h vector of each day is substituted by a cluster number; therefore, the real valued time series of $\{\boldsymbol{E}(24), \boldsymbol{E}(48), \ldots, \boldsymbol{E}(n-24), \boldsymbol{E}(n)\}$ is replaced with an integer valued time series of $\{c(1), c(2), \ldots, c(\frac{n-24}{24}), c(\frac{n}{24})\}$ where $c(\frac{i}{24})$ indicates the cluster label for data point $\boldsymbol{E}(i)$. Now, in order to find $\widehat{\boldsymbol{E}}(t_{s^*})$, where $t_{s^*} \in t_s$ is an instance of test set indices, a template of cluster labels for the previous days $\{c(t_{s^*}-D), \ldots, c(t_{s^*}-2), c(t_{s^*}-1)\}$ is created where similar to kNN, D is the depth of comparison (which is called window length in [28]). Then, the time series of all the preceding days of t_{s^*}, i.e. $\{c(1), c(2), \ldots, c(t_{s^*}-2), c(t_{s^*}-1)\}$, is matched against the above mentioned template. $\widehat{\boldsymbol{E}}(t_{s^*})$ is then equal to the corresponding cluster center of the day following immediately after the most recent matched template index. If there is no match for the template with the length of D, the template length is shortened to $D-1$ and algorithm iterates until there is some match in the time series. Similar to kNN, parameter D is determined through cross validation.

The steps of MPSF are detailed in Fig. 3.

TWDP based Nearest Neighbor Algorithm

Inputs: $\boldsymbol{x}(t_r), \boldsymbol{y}(t_r), \boldsymbol{x}(t_{s^*})$
Output: $\boldsymbol{y}(t_{s^*})$
1. **for** $j \in t_r$
2. $\quad \boldsymbol{dis}[j] = -\boldsymbol{x}^t(t_{s^*}).diag(\boldsymbol{TW}).\boldsymbol{x}(j)$
3. $idx = \arg(\min_j(\boldsymbol{dis}[j]))$
4. $\boldsymbol{y}(t_{s^*}) = \boldsymbol{y}(idx)$

Fig. 2. Nearest Neighbor Algorithm with TWDP similarity measure.

Modified Pattern Sequence-based Forecasting (MPSF) Algorithm

Inputs: $\{\boldsymbol{E}(t_r)\} = \{\boldsymbol{E}(24), \boldsymbol{E}(48), \ldots, \boldsymbol{E}(n-24), \boldsymbol{E}(n)\}, D$
Output: $\widehat{\boldsymbol{E}}(t_{s^*})$
1. **for** $k \in \{0.1 \times |unique\{\boldsymbol{E}(t_r)\}|, \ldots, |unique\{\boldsymbol{E}(t_r)\}|\}$
2. Perform k-means clustering with $k \rightarrow C_k$ with cluster centers $\boldsymbol{CC}_1, \boldsymbol{CC}_2, \ldots, \boldsymbol{CC}_k$
3. Calculate $SI[k]$
4. $k^* = \arg(\min_k SI[k])$
5. Replace $\{\boldsymbol{E}(t_r)\}$ with $\{c(\frac{t_r}{24})\}$ according to C_{k^*}
6. **while** $(idx = \emptyset \;\&\; D > 0)$
7. $\boldsymbol{temp}(t_{s^*}) = \{c(t_{s^*} - D), \ldots, c(t_{s^*} - 2), c(t_{s^*} - 1)\}$
8. $idx = \max_i(find(\boldsymbol{temp}_{i \in t_r}(i) == \boldsymbol{temp}(t_{s^*})))$
9. $D = D - 1$
10. **if** $D = 0$
11. $idx = \arg(\max_i(\text{count}_{i \in t_r} c(i)))$
12. $\widehat{\boldsymbol{E}}(t_{s^*}) = \boldsymbol{CC}_{c(idx)}$

Fig. 3. MPSF algorithm according to [29].

4.3. Support Vector Regression (SVR)

SVR is an expansion of the Support Vector Machines (SVMs) idea from classification to regression [30]. The idea is that for classification, there is no need to use all training samples to construct the decision boundaries, rather a few samples, called Support Vectors, are important. The ε-SV regression algorithm is one of the extensions of SVMs to regression problems. For our problem here, the ε-SV formulation can be defined as following:

$$\widehat{\boldsymbol{E}}(i) = f(\boldsymbol{E}(i-1)) = \langle \boldsymbol{E}(i-1), \boldsymbol{\omega} \rangle + b, \tag{3}$$

where $\boldsymbol{\omega} \in \{\boldsymbol{E}(t_r)\}$ and $b \in \mathcal{R}$ are the solutions to the following optimization problem:

$$\begin{aligned} &\text{minimize} \frac{1}{2}\|\boldsymbol{\omega}\|^2 + C\sum_{i=1}^{N_{tr}}\sigma_i \\ &\text{subject to} \begin{cases} |\langle \boldsymbol{E}(i-1), \boldsymbol{\omega} \rangle + b - E(i)| \leqslant \varepsilon + \sigma_i \\ \sigma_i \geqslant 0 \end{cases} \end{aligned} \tag{4}$$

It is not always possible to find $\boldsymbol{\omega}, b$ such that all the $E(i)$ and $\widehat{E}(i)$ lie in ε distance of each other. By adding slack variables σ_i and the coefficient C, $\boldsymbol{\omega}$ is obtained as follows [30]:

$$\boldsymbol{\omega} = \sum_{i=1}^{N_{tr}}(\alpha_i - \alpha_i^*)\boldsymbol{E}(i-1), \tag{5}$$

where $\alpha_i,\ \alpha_i^*$ are Lagrange multipliers. Note that $\alpha_i,\ \alpha_i^*$ are nonzero only for training samples that violate the ε proximity constraint. Therefore, $\boldsymbol{\omega}$ only depends on a few training samples that are in fact support vectors.

Thus far, f was a linear function of the training samples, but it can be extended to include non-linear functions of the training samples through kernel trick [30]:

$$f(\boldsymbol{E}(j-1)) = \sum_{i=1}^{N_{tr}}(\alpha_i - \alpha_i^*)G(\boldsymbol{E}(i-1), \boldsymbol{E}(j-1)) + b \tag{6}$$

where $G(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is a kernel function. Example of popular kernels are polynomial, $G(\boldsymbol{x}_i, \boldsymbol{x}_j) = (\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + c)^p$, hyperbolic tangent, $G(\boldsymbol{x}_i, \boldsymbol{x}_j) = \tanh(a\langle \boldsymbol{x}_i, \boldsymbol{x}_j \rangle + c)$ (for some positive a), and Gaussian radial basis function, $G(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(-\gamma\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2\right)$ for $\gamma > 0$.

In this paper we used function 'svm' in the package 'e1071' of the R programming language.

4.4. Random Forest (RF)

The Random Forest algorithm is an aggregated version of decision trees [31]. A decision tree tries to learn a set of rules to predict the output value for an unseen input. To this end, at each node, a decision criterion is defined. For instance, in our application, the first rule (corresponding to the root node of the tree) could be that if power consumption is less than 1 kW, the hour ahead consumption will be 1.2 kW. Most probably this rule is not enough for precise prediction, so another rule (corresponding to another node) will be added which creates the new rule: if power consumption is less than 1 kW and last hour's consumption is more than 0.75 kW, then the hour ahead consumption is 1.4 kW. This process will keep going until there is no undecided input variable or the desired precision has been met or other stopping criteria such as maximum number of terminal nodes for the tree has been observed. At each node, selecting the input variable to split is usually done to optimize some criteria. Two common criteria are entropy and Gini impurity [32]. For instance, if we use the entropy criteria, at each node, we will split an input variable that reduces the uncertainty as much as possible. In other words, we will pick a variable to split that gives us the most certain decision.

Decision trees suffer from overfitting; that is, they are high variance models for data. This problem leads to decision trees having poor generalization and hence poor prediction accuracy. One way to solve this problem is using Random Forest (RF) algorithm. RFs do not use just one tree but many decision trees where the training set of each tree is made of N_{tr} training data points, sampled with replacement from the original training set of size N_{tr}. Also, when finding a variable to split at each node of each tree, instead of all input variables, we look at a random subset of variables. Consequently, Random Forest is a randomized and forested version of the decision trees.

RF has been shown to be very powerful in classification and regression problems [33]; in this paper we have used function 'randomForest' in the package 'randomForest' of the R programming language.

5. Data and preprocessing

The prediction algorithms described in the next section are applied to the charging stations located on the UCLA campus. The data used in this paper were recorded from December 7, 2011 to February 28, 2014; however, not all outlets were installed at the same time or were in use on all days. Among the charging outlets at UCLA, 28 outlets have charging data for more than 30 effective days (days that nonzero charging has been reported); the data from these outlets have been used in this paper.

Data for each outlet comes in two formats: Charging Records and Station Records. We are interested in analyzing the difference in using prediction algorithms on each of these formats. These formats have been explained below, as well as the timing and stages of pre-process for each format in order to acquire the time series.

5.1. Station records

This dataset comes directly from measurements at the outlets. Thus, when accessing this dataset from the server, it is not part of the customer's profile; rather it is the recorded quantities at the outlet. Each station record contains measurements, such as voltage, current, and power factor of the charging outlet, in three to five minute intervals. In order to obtain the real power time series, we multiply the voltage, current, and power factor.

However, not all of these quantities are available or have been reported correctly at all instances. In this paper, we classify data which is either unreported or the data that is not in the nominal expected range (and hence non usable) as missing data.

The process of providing the best guess for the missing values is called imputation [34]. Some imputation methods involving deletion, such as "case deletion" where the incomplete instance of data is removed from the dataset, are not suitable for time series since they will change the relative order of events and make the time series lose its ordinal properties such as periodicity. A more elaborate discussion on imputation methods and their application on energy time series have been discussed in [35]. The article argues that each prediction algorithm goes along well with a certain imputation method and care should be taken in selecting an imputation method for each prediction algorithm. Based on this reference, we chose zero imputation for SVR and MPSF prediction algorithms and median imputation for RF and NN prediction algorithms.

In the case of zero imputation, the value of zero is substituted for all missing values. In this method, missing voltage, current, or power factor, is substituted with zero. This imputation preserves the sparsity of the time series. In median imputation, on the other hand, missing values of each quantity are being replaced with the median of that quantity for that specific outlet. The advantage of the median imputation method is that the imputed value is always one of the actual values of the data.

Before applying the imputation methods, missing values need to be identified. For voltage, current, and power factor, in addition to unreported and negative reported values, the reported values that were more than the maximum voltage, maximum current and maximum power factor (one) were identified as missing values.

Since we are interested in comparing the prediction results with that of the charging records dataset and hence forecast on hourly basis, we upsample the time series to form a power time series of one hour granularity for each outlet.

5.2. Charging records

This dataset comes from anonymous user profiles. Every time that an EV uses a charging facility, a charging record is added to the EV profile. Each charging record contains the charging time (beginning and end) and the acquired energy in kWh. In order to make the times series with one hour granularity, the Charging Records are converted to time series by uniformly dividing the acquired energy to the (rounded) charging interval; e.g., if the charging interval is 3.2 h and the acquired energy is 3 kW h, it is assumed that the EV received 1 kW h of energy in each hour.

Charging records are different from station records in that missing data cannot be easily identified. This is because charging records are event triggered measurements. In the case of station records, when we do not receive a value (i.e. for voltage) in a five minute period, we know that the value is missing; however, if we do not receive any charging record in a time period, we conveniently assume no charging event has occurred. Therefore, there is no need for missing value imputation in charging record dataset.

5.3. Comparing two datasets

As explained earlier, the main difference between the two datasets is that charging records are derived from user profile data. These records include the entrance and exit time of each EV and hence are prone to jeopardizing the user's privacy and could lead to misuse. Station records, on the other hand, are direct measurements of quantitates at the outlet and are independent from customer or particular EV information.

The figure below shows constructed time series with one hour granularity (as explained above) from both dataset formats for a sample day (August 13, 2013, to be specific) for one of the outlets.

According to Fig. 4, since the charging record (the diamond blue curve) indicates the beginning and ending of the EV presence at the outlet, we can speculate that one EV has been present at the outlet from 8 to 18 while another EV has been present from 18 to 20. The station record (the diamond orange curve), however, shows the time when the EV is actually receiving power from outlet, which for the first EV is from 9 to 15 and for the second one only one hour at 20. So these two datasets are describing the same phenomenon with different accuracy. It is important to note that the areas under both curves are equal to each other, meaning that they both report the same amount of energy being consumed. If the ultimate goal is predicting the available energy at each outlet as in [36], the predictions based on either one are expected to behave competitively.

Another difference between these two types of datasets is their preprocessing time. Each charging record is made of three values (beginning and end of charging and acquired energy), and the time required for preprocessing is divided between accessing the data base and creating the time series from those three values. However, for preparing the time series from station records, one needs to access the database, identify missing values and impute them and, finally, upsample the time series to achieve the time series with one hour granularity. Therefore, it should not be surprising that preprocessing time for station records takes more than charging records.

Fig. 5 shows the preprocessing time for charging record dataset and station record dataset per each outlet.

Depending on the number of charging records for each outlet, the preprocessing time will be different. Note that for a certain outlet, the preprocessing time will not change by the rate of usage of the outlet, while charging records dataset is a direct function of the number of charging events for the outlet. If there is no charging event for an outlet, then there will be no charging record and the time series is readily available, but for a station record all the steps of identifying missing values and possibly imputation as well as upsampling should be performed.

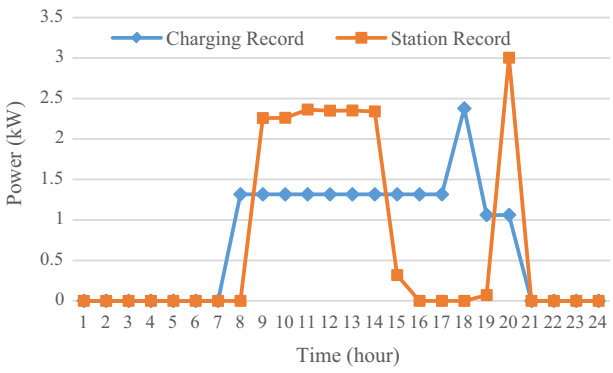

Fig. 4. Time series constructed from station record and charging record formats.

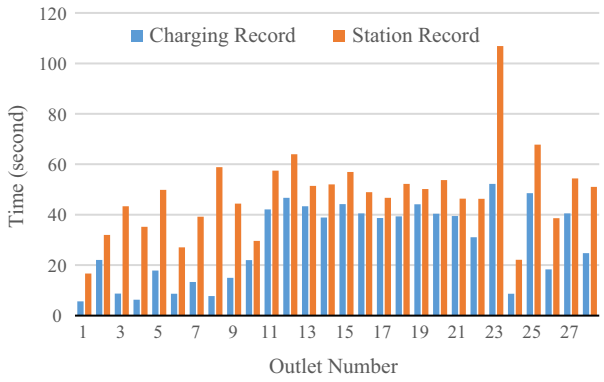

Fig. 5. Preprocessing times for preparing times series with both format of data per outlet.

Thus, we see that one difference between these two datasets is that it always takes longer to preprocess (and eventually predict) for station records compared with charging records.

6. Simulations and analysis

In this section we first describe our parameter selection procedure, followed by reporting the results and analyzing them.

6.1. Parameter selection

The following combinatorial parameters need to be determined for our algorithms via cross validation: Depth (D) for all algorithms, minimum number of terminal nodes, number of trees, and number of variables randomly sampled at each split for RF algorithm. Also for the ε-SV, the tradeoff coefficient C, desired ε, kernel type, and its corresponding parameters need to be determined via cross validation.

There are some challenges with cross validation when applying machine learning methods to time series forecast problems [37,38]. These challenges have been pointed out in [29], and a modified form of the blocked cross-validation has been chosen since it seems to possess advantages mentioned in both machine learning and time series forecasting literature.

Blocked cross validation [29] is similar to k-fold cross validation, except that the order of the samples in each block is preserved. Now, since the first block does not have any preceding values, the modification is to select cross validation blocks after a minimum training data (which is used for training the first block). Fig. 6 illustrates the modified blocked cross validation with five validation blocks. First, the algorithm is trained on {T1,T2} blocks and validated on the V1 block; then, it is trained on {T1,T2,V1} blocks and validated on the V2 block, and so on. This cross validation method uses the maximum possible data (in comparison with last block validation) while respecting the temporal order of time series data.

In cross validation, the depth parameter (D) varies between 1 and 30 (equal to looking only at yesterday and up to the past month). The kernel type for SVR was selected from linear, radial basis, sigmoid, and polynomial. Other parameters for SVR and their ranges are (the bold parameter is the default in the relevant R package): $\varepsilon \in \{0.01, \mathbf{0.1}\}$ and $C = \{0.1, \mathbf{1}\}$. Similarly for RF, the parameters are the number of trees, $nt \in \{200, \mathbf{500}\}$, number of variables to consider for splitting at each node, $m \in \{\frac{1}{6}, \frac{\mathbf{1}}{\mathbf{3}}, \frac{2}{3}\} \times D$, and minimum size of terminal nodes, $ns \in \{5, 10\}$. There are lots of other parameters for SVR and RF for which we used their default value in the relevant R package.

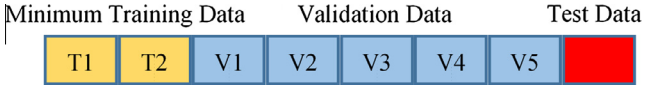

Fig. 6. Modified blocked cross validation. Training data is divided to minimum training data {T1,T2} and validation data {V1,...V5}. Model is first trained on minimum training data {T1,T2} and evaluated on V1, then it is trained on {T1,T2,V1} and evaluated on V2, up until training on {T1,T2,V1,...,V4} and evaluating on V5.

6.2. Results

The training set in our simulations was the first 90% of the data which makes the test set the last 10% of the data. We used five blocks in the cross validation procedure.

Table 1 shows the average prediction SMAPE on all outlets for each algorithm when using either charging record or station record time series.

Care should be taken before deciding which source of data leads to a better prediction accuracy based on just the average accuracy over all the outlets, due to varied performance of each algorithm on each individual outlet. In making such a decision, statistical analysis needs to be considered.

Fig. 7 shows the average SMAPE on test days based on both datasets per prediction algorithm and outlet. In the analysis section we will investigate whether there is a statistically significant difference in the accuracy of the algorithms when using either of the datasets.

Note that since time series attained from either of datasets have the same granularity, the processing time for a certain prediction algorithm and certain outlet on either of time series would be the same.

The analysis has been performed with RStudio version 0.98.1091 and Microsoft SQL Server Management Studio on an Intel Core i-7 CPU at 3.40 GHz with 16 GB RAM. RStudio is running under R version 3.1.2.

6.3. Analysis

There are various criteria to judge whether a set of results is better than another. In general, the criterion depends on the application. For example in a business model, if the penalty depends on average SMAPE, the algorithm with less average SMAPE will be selected. However, if we want to know how often it is probable that two set of results significantly vary from each other, we need to use statistical tests.

In this section, we want to see whether there is a statistically significant difference between the accuracy of the predictions based on either of the datasets. To this end, we use the Wilcoxon signed rank test [39]. In our application, this test compares the SMAPE resulted from two different datasets at each outlet. Depending on the number of positive and negative differences and their absolute value, the test gives a probability (p-value) of how likely these two SMAPE results come from the same distribution. A small p-value means that it is unlikely that the two SMAPE population come from the same distribution, and hence there is a statistically significant difference between the results.

Table 1
Average of smape (%) on test days for charging record and station record based time series for each algorithm.

Time series origin	NN-TWDP	MPSF	SVR	RF
Charging record	10.02	7.85	19.79	20.82
Station record	16.45	6.28	20.68	20.09

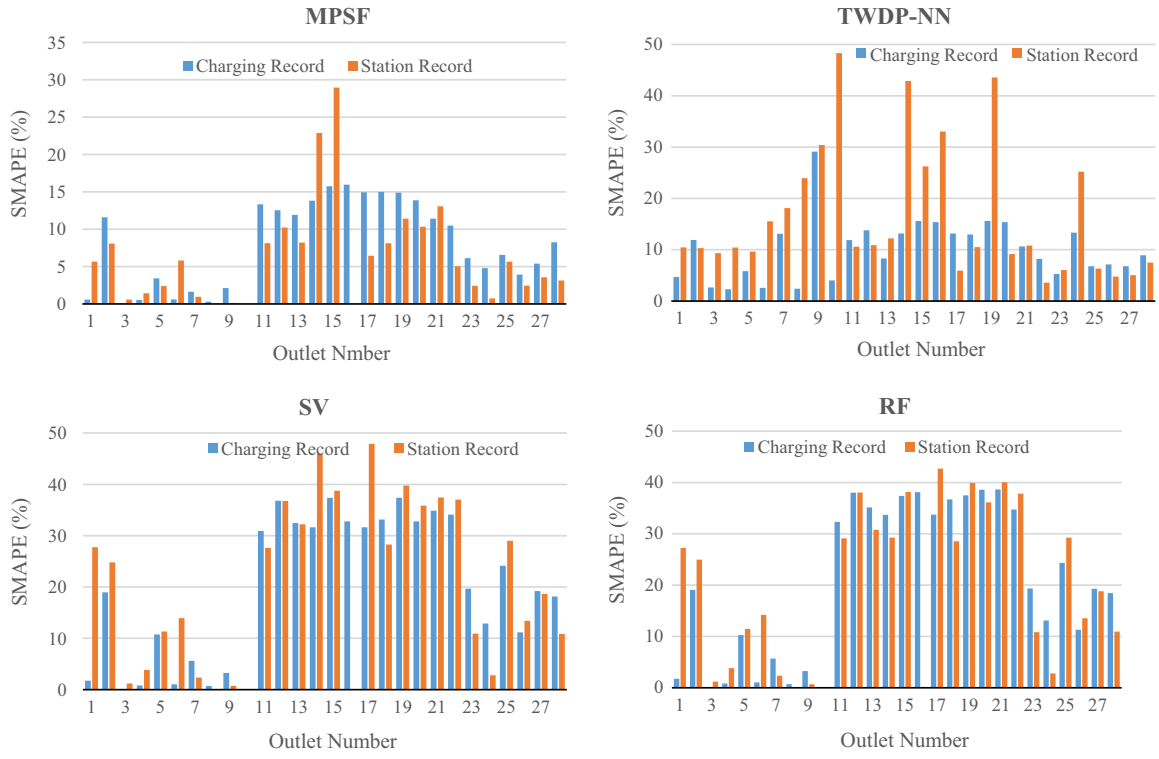

Fig. 7. Average Symmetric Mean Absolute Percentage Error (SMAPE) on test days based on both datasets for each outlet for TWDP-NN, MPSF, SVR, and RF algorithms.

It is common to pick a threshold for the p-value which is called significance level and denoted by α. It is customary to pick α equal to 0.01, 0.05 or 0.1 [39]. We pick $\alpha = 0.05$. Table 2 shows the p-value for applying the Wilcoxon test on each of the prediction algorithm results depicted in Fig. 7.

By selecting the significance level (α) of 0.05 for p-value, there is no statistically significant difference between using charging record and station record based time series for SVR and RF algorithms (corresponding p-values are greater than 0.05). However, since the p-value is less than 0.05 for NN-TWDP and MPSF algorithms, there is a statistically significant difference between prediction results when using charging record or station record time series.

For NN-TWDP and MPSF, where the statistically significant difference between results has been observed, the dataset with greater occurrence of lower SMAPE is considered preferable. For NN-TWDP, most of the outlets have lower SMAPE when using the charging record; while for MPSF, most of the outlets have lower SMAPE when using the charging record (Fig. 7). Therefore, statistically speaking, the charging record based time series gives higher accuracy when NN-TWDP is the prediction algorithm; however, the station record based time series gives higher accuracy when MPSF is the prediction algorithm. There is no difference between using either of the datasets for RF and SV algorithms.

Next, we investigate whether there is a statistically significant difference between the prediction results of the two datasets, regardless of the algorithm. In order to make an overall conclusion on the effect of these two datasets on the prediction accuracy, one approach is to perform the Wilcoxon signed rank test on all the results together. This way instead of 28 samples (number of outlets) per algorithm, we have 112 samples (28 outlet and four algorithms) for all four algorithms. In this case, the p-value from the Wilcoxon test is 0.6136 which shows that, overall, there is no statistically significant difference in using either of the charging record or station record based time series. This conclusion is not unexpected since the charging records are essentially zeroth-order approximation of the station records dataset, and the station records themselves are usually constant on pretty large intervals (e.g. Fig. 4). We speculate if the station records were a time series with lots of fluctuations, charging records, as it is zeroth-order approximation, would result in significantly worse prediction accuracy.

Table 2
p-values of Wilcoxon signed rank test on charging record and station record based time series for each algorithm.

	NN-TWDP	MPSF	SVR	RF
p-value	0.02983	0.04235	0.4785	0.7639

Table 3
Summary of differences between prediction with charging record and station record based time series.

Time series origin	Speed	Prediction error (SMAPE)	Privacy preserving
Charging record	Preprocessing on average twice as fast	Lower SMAPE for NN-TWDP (Generally no statistically significant difference)	No (comes from customer profile)
Station record	Preprocessing on average twice as slow	Lower SMAPE for MPSF (Generally no statistically significant difference)	Yes (independent from each particular customer)

7. Conclusion

In this paper, we investigated the difference between prediction based on time series obtained from charging records and station

records. A charging record contains three values for each charging event (beginning and end of charging and the acquired energy) which comes from customer profiles. A station record, on the other hand, comes from station measurements and is a five-minute log of voltage, current, and power factor and its length depends on the length of the charging event.

Because of the greater volume of data per charging event for station records, preparing the time series from station record dataset takes longer than charging record dataset; hence, for fast prediction applications charging record is more suitable. On the other hand, charging records are part of the customer profile (although anonymous) and yields privacy concerns while station records come from station measurement without any access to customer behavior.

Perhaps the more important question is the difference in the prediction error when using these two datasets. In general, there is no statistically significant difference between prediction errors, although looking at the results for each algorithm demonstrates that for NN and MPSF, the charging records and station records create less error respectively.

Table 3 summarizes the difference between the two datasets when used for prediction. The table can guide an application designer to choose the right dataset depending on the speed and privacy concerns as well as prediction algorithm used for the application in hand.

Acknowledgment

This work has been sponsored in part by grant from the LADWP/DOE fund 20699 & 20686, Smart Grid Regional Demonstration Project.

References

[1] McDaniel P, McLaughlin S. Security and privacy challenges in the smart grid. IEEE Secur Priv 2009;7(3):75–7.
[2] Majidpour M, Qiu C, Chu P, Gadh R, Pota H. A novel forecasting algorithm for electric vehicle charging stations. In: Proc. 2014 Intl. Conf. Connected Vehicles and Expo (ICCVE).
[3] Sweeney L. Weaving technology and policy together to maintain confidentiality. J. Law. Med. Ethics 1997;25(2–3):98–110.
[4] Krishnamurti T, Schwartz D, Davis A, Fischhoff B, de Bruin WB, Lave L, et al. Preparing for smart grid technologies: a behavioral decision research approach to understanding consumer expectations about smart meters. Energy Policy February 2012;41:790–7.
[5] Bae M, Kim H, Kim E, Chung AY, Kim H, Roh JH. Toward electricity retail competition: survey and case study on technical infrastructure for advanced electricity market system. Appl Energy November 2014;133:252–73.
[6] McKenna E, Richardson I, Thomson M. Smart meter data: balancing consumer privacy concerns with legitimate applications. Energy Policy February 2012;41:807–14.
[7] Iversen EB, Morales JM, Madsen H. Optimal charging of an electric vehicle using a Markov decision process. Appl Energy June 2014;123:1–12.
[8] Mathew PA, Dunn LN, Sohn MD, Mercado A, Custudio C, Walter T. Big-data for building energy performance: lessons from assembling a very large national database of building energy use. Appl Energy February 2015;140:85–93.
[9] Xu Z, Hu Z, Song Y, Zhao W, Zhang Y. Coordination of PEVs charging across multiple aggregators. Appl Energy December 2014;136:582–9.
[10] Rottondi C, Verticale G, Krauss C. Distributed privacy-preserving aggregation of metering data in smart grids. IEEE J Sel Areas Commun 2013;31(7):1342–54.
[11] Rial A, Danezis G. Privacy-preserving smart metering. In: Proceedings of the 10th annual ACM workshop on privacy in the electronic society. New York, NY, USA; 2011. P. 49–60.
[12] Erkin Z, Troncoso-Pastoriza JR, Lagendijk RL, Perez-Gonzalez F. Privacy-preserving data aggregation in smart metering systems: an overview. IEEE Signal Process Mag 2013;30(2):75–86.
[13] Kalogridis G, Denic SZ. Data mining and privacy of personal behaviour types in smart grid. In: 2011 IEEE 11th international conference on data mining workshops (ICDMW); 2011. P. 636–42.
[14] Kalogridis G, Cepeda R, Denic SZ, Lewis T, Efthymiou C. ElecPrivacy: evaluating the privacy protection of electricity management algorithms. IEEE Trans Smart Grid 2011;2(4):750–8.
[15] Lindell Y, Pinkas B. Privacy preserving data mining. In: Bellare M, editor. Advances in cryptology — CRYPTO 2000. Berlin Heidelberg: Springer; 2000. p. 36–54.
[16] Agrawal R, Srikant R. Privacy-preserving data mining. In: Proceedings of the 2000 ACM SIGMOD international conference on management of data. New York, NY, USA; 2000. P. 439–50.
[17] Shaneck M, Kim Y, Kumar V. Privacy preserving nearest neighbor search. In: Machine learning in cyber trust. Springer US; 2009. P. 247–76.
[18] Yang Z, Zhong S, Wright RN. Privacy-preserving classification of customer data without loss of accuracy. In: SIAM SDM; 2005. P. 21–23.
[19] Yu H, Jiang X, Vaidya J. Privacy-preserving SVM using nonlinear kernels on horizontally partitioned data. In: Proceedings of the 2006 ACM symposium on applied computing. New York, NY, USA; 2006. P. 603–10.
[20] Chaudhuri K, Monteleoni C. Privacy-preserving logistic regression. In: Koller D, Schuurmans D, Bengio Y, Bottou L, editors. Advances in neural information processing systems, vol. 21. Curran Associates, Inc.; 2009. p. 289–96.
[21] De Gooijer JG, Hyndman RJ. 25 years of time series forecasting. Int J Forecast 2006;22(3):443–73.
[22] Ahmed NK, Atiya AF, Gayar NE, El-shishiny H. An empirical comparison of machine learning models for time series forecasting.
[23] Bishop CM. Pattern recognition and machine learning. Springer; 2006.
[24] Hastie T, Tibshirani R, Friedman J. The elements of statistical learning: data mining, inference, and prediction. second ed. Springer; 2009.
[25] 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;11(1):242–50.
[26] Dietterich TG. Machine learning for sequential data: a review. In: Structural, syntactic, and statistical pattern recognition. Springer; 2002. p. 15–30.
[27] Martínez-Álvarez F, Troncoso A, Riquelme JC, Aguilar-Ruiz JS. LBF: a labeled-based forecasting algorithm and its application to electricity price time series. In: Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on; 2008. P. 453–61.
[28] Martinez Alvarez F, Troncoso A, Riquelme JC, Aguilar Ruiz JS. Energy time series forecasting based on pattern sequence similarity. Knowl Data Eng IEEE Trans 2011;23(8):1230–43.
[29] Majidpour M, Qiu C, Chu P, Gadh R, Pota HR. Modified pattern sequence-based forecasting for electric vehicle charging stations. In: 2014 IEEE international conference on smart grid communications (SmartGridComm); 2014. P. 710–15.
[30] Smola AJ, Schölkopf B. A tutorial on support vector regression. Stat Comput 2004;14(3):199–222.
[31] Breiman L. Random forests. Mach Learn 2001;45(1):5–32.
[32] Breiman L. Technical note: some properties of splitting criteria. Mach Learn 1996;24(1):41–7.
[33] Caruana R, Niculescu-Mizil A. An empirical comparison of supervised learning algorithms. In: Proceedings of the 23rd international conference on machine learning; 2006. P. 161–68.
[34] Allison PD. Missing data. SAGE Publications; 2001.
[35] Majidpour M, Qiu C, Chu P, Gadh R, Pota H. Treatment of missing values in electric vehicle charging load prediction. IEEE Trans. on Smart Grid [submitted for publication].
[36] Majidpour M, Qiu C, Chung C-Y, Chu P, Gadh R, Pota HR. Fast demand forecast of electric vehicle charging stations for cell phone application. In: 2014 IEEE PES general meeting|conference exposition; 2014. P. 1–5.
[37] Bergmeir C, Benítez JM. On the use of cross-validation for time series predictor evaluation. Inf Sci 2012;191:192–213.
[38] Opsomer J, Wang Y, Yang Y. Nonparametric regression with correlated errors. Stat Sci 2001:134–53.
[39] Derrac J, García S, Molina D, Herrera F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol Comput 2011;1(1):3–18.