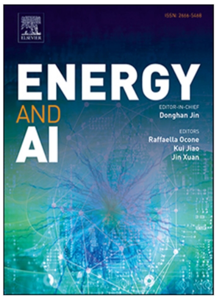

A data-driven framework for medium-term electric vehicle charging demand forecasting

Alexander Orzechowski, Loren Lugosch, Hao Shu, Raymond Yang, Wei Li, Brett H. Meyer

PII: S2666-5468(23)00039-3
DOI: https://doi.org/10.1016/j.egyai.2023.100267
Reference: EGYAI 100267

To appear in: *Energy and AI*

Received date : 20 January 2023
Revised date : 29 March 2023
Accepted date : 18 April 2023

A Data-Driven Framework for Medium-Term Electric Vehicle Charging Demand Forecasting

Alexander Orzechowski[a], Loren Lugosch[a], Hao Shu[a], Raymond Yang[a], Wei Li[b], Brett H Meyer[a]

[a]*McGill University, Mcconnell Engineering Building, 3380 University St, Montreal, QC H2X 2G6, Canada*
[b]*Mogile Technologies, 16647 Hymus Blvd, Kirkland, Quebec H9H 4R9, Canada*

Abstract

The rapid phase-in of electric vehicles (EV) will cause unprecedented issues with managing the supply of electricity and charging stations. It is in the interest of utility providers and everyday consumers to be able to plan for peak charging times, and related congestion. While past work has been done for localized, short-term forecasting, it has not included longer term forecasting, or considered the relationships between multiple stations. Importantly, past work has also not offered a framework for dataset construction and evaluated different dataset features. We propose a methodology to forecast demand at public EV charging stations, and use it to explore the potential of data-driven models to predict demand up to one week in advance. Our strategy includes selecting parameters for formatting a dataset given a list of charging events, a way to consider flexible prediction horizons, and deployment of deep and supervised learning-based models. To the best of our knowledge, ours is the first study to propose machine learning to forecast medium-term public EV charging demand, to exploit weather and other features at public charging stations, and to forecast demand at multiple stations and the entire network. We validated our approach using data from eleven stations over three years from Scotland, UK. Our method outperforms the benchmark time series method, and predicts network demand with a symmetric mean absolute percentage error (SMAPE) of 5.9% and a mean absolute error (MAE) of 124.7kWh, or less than twelve percent of average daily demand.

Keywords: Public EV Charging, Forecasting, Energy Consumption, Smart Grid, Time-Series, Multi-Task Learning, Deep Learning

1. Introduction

Electric Vehicles (EVs) will be deployed in large scale in the next one to two decades [1], [2]. However, the electric grid and cities are not configured for widespread EV use, so changes are needed in a wide array of domains. Power grid infrastructure is being upgraded [3]. Vehicle-to-grid and vehicle-to-building (V2G/V2B) technology is being studied to allow for grid flexibility. At the same time, utilities providers are expected to implement pricing models to incentivize customers to charge at non-peak times. Consequently, consumers and utilities alike would benefit from demand forecasting: consumers, to avoid wait times at charging stations; utilities, to address short- to long-term power needs.

The ability to predict demand at public charging stations helps providers avoid higher costs of energy and excessive risks to grid disruption. Prediction can also help to alleviate traffic congestion, and decision making when EVs participate in the electricity trading market [4]. Additionally, the scarcity of charging resources, and waiting times, are expected to worsen due to increasing public charging demand [5]. Demand prediction can further improve the operational efficiency of utilities providers because supplemental plants may need to be brought online during peak load. Importantly, this complex problem of coordinating EV charging also involves scheduling algorithms that can benefit from machine learning demand predictions [6].

Our objective is to build a framework to process and make use of the charging events collected by utilities providers. Then prediction models, combined with the framework, are deployed to predict the demand at various stations in a charging network, as well as for the network as a whole, to support both consumers and utilities. Given past charging sessions, the model will predict future usage up to seven days in advance. Utilities providers will need to be able to dynamically adjust the pricing, and possibly power, of charging stations to influence when and where drivers charge their cars in order to not overload the electrical grid. In the meantime, drivers may wish to adjust their plans to avoid congestion, and therefore reduce waiting time. Our modeling approach addresses these needs.

Predicting public charging behavior is challenging for a few reasons. The first is because of the random nature of EV charging in a public charging station environment. Specifically, Figure 1 (d) shows the distribution of energy values requested at a public charging stations, where energy demand varies from 1kWh to 40kWh (approximately 5km to 200km of driving range). Second, there is little consensus on the appropriate features to use to predict demand. Some studies rely on personal information. Furthermore, features such as holidays, weather, and activity at nearby stations have been rarely studied, despite their statistical significance [7]. There is also disagreement about whether energy, power, waiting time, or charging time, are the most appropriate metric to predict, as well as over what period of time [4], [7].

Load forecasting in power systems is well established and has been characterized using a few different time horizons: short-, medium-, or long-term. Short-term load forecasting (STLF) includes a half-hour to 24 hours ahead. Medium-term (MTLF) ranges from one day to one year in advance, while long-term load forecasting (LTLF) is from one to ten years [8]. Forecasting in a power grid is concerned with supporting decision making at the granularity of hourly, daily, weekly, and monthly changes in load, according to Gross and Galiana [9]. Load can be defined as the system load, peak system load, and the system energy [8]; we focus on predicting system energy demand at daily granularity.

Prior research has made progress on the challenge of forecasting for public charging stations. For example, in some studies data was for a car sharing platform, with data associated to registered users, meaning user-specific data could be leveraged. In contrast, others were for residential or university campus data. Overall, these works generally study short term demand [10, 11]. Our work extends to the medium-term demand for energy at public charging stations.

Previous works have primarily used machine learning (ML), and more rarely statistical methods. Most ML studies have used only past data, for example using past power consumption data to predict future power consumption, as opposed to exploring a variety of relevant features [4, 12]. Some works have relied on traffic to predict demand, or tracking user driving behavior and attributes of drivers' vehicles. The purpose of these studies is generally to predict EV usage and schedule EVs, for a fleet of vehicles, as opposed to for public charging stations [4, 13, 14]. Overall, the related studies do not address the problem sufficiently because they did not consider the data at public charging stations, a prediction horizon for medium to long-term forecasting, or a study of relevant features or algorithms for forming a feature engineering framework.

Lastly, previous research has been limited by the assumptions made the about historical data used to train models. Specifically, some work used simulated data. However, studies with real data are important because simulated charging data cannot capture all the relationships and stochastic effects of real behavior [3]. Methods for modelling demand at public EV charging stations rarely appear in the literature, primarily due to a lack of data from utilities [3]. Additionally, some studies assumed the availability of user-level behavioral data. While such data makes it easier to answer certain questions, trends toward protecting user privacy means that such data may not generally be available. Consequently, our study focuses less on the question of when will an outlet currently being used be free, and more on what will be the aggregate demand at charging stations and in networks, since the former is extremely stochastic in nature and has only been studied with user behavior data [13, 15].

In this paper, we propose a robust ML pipeline that, given anonymized public charging station usage data, (a) constructs and labels a dataset consisting of daily station energy usage, (b) extends it using other relevant features, and (c) explores the impact of different time horizon parameters as well as different output layer structures. We evaluate machine learning and other methods for the prediction of daily demand at individual stations and in the whole network. We also study the potential of multi-task learning (MTL), and observe that when a single model predicts demand for individual stations as well as the entire network, accuracy improves. Using a dataset with over 66,000 charging events, an Artificial Neural Network (ANN) model achieves a symmetric mean absolute percentage error (SMAPE) of 5.9% for the network, and 21.3% for individual stations, on average.

The remainder of this paper is organized as follows. Section 2 summarizes the related existing literature and further highlights this study's contributions with respect to the literature gaps. Section 3 gives an overview of the charging event dataset used in this study. Section 4 presents our data-driven framework for charging demand forecasting, along with experiments used to choose training dataset parameters. Section 5 presents the setup for our experimental evaluation of our framework, and details the various techniques we compare it with. Section 6 reports and discusses the results of our evaluation; and, Section 7 concludes with a brief summary of our methods and findings.

2. Related Work

In the past seven years, there have been several studies on the application of forecasting EV charging station load. A few have focused on public EV charging stations. The study in [11] employed machine learning with k-nearest neighbor (KNN) to predict the aggregate demand up to one day in advance at a few EV charging outlets on a university campus. The same university campus was studied in [16], which proposed a KNN combined with a new method known as pattern sequence-based forecasting (PSF). PSF is a recently proposed time-series forecasting algorithm that found success in energy price forecasting. Similarly, the work in [17] also studied the demand for the next day with hourly time intervals and examined a few models moving towards an ensemble learning method. The benefit of ensemble learning was 1% to 8% depending on error metric, Root Mean Square Error (RMSE) or Mean Absolute Error (MAE), compared to a single model approach. Their method was demonstrated for predicting aggregated network demand for one city in Colorado. Overall, these studies [11, 16, 17] only study demand for the next 24 hours at a single geographical location and do not consider non-demand based features: the models rely on the assumption that only past charging demand causes future charging demand.

More recently, Ye et al. in [18] studied the short term demand for public charging stations near freeways, with limited charging data available (48 days). Their study included Random Forest, ANN, and LSTM with a few interesting features including the real freeway traffic flow. Also recently, the study by Shahriar et al. [6] demonstrated the use of features to predict session duration and energy consumption for a single charging event. The ensemble model proposed achieves a SMAPE of about 11% for event energy. In contrast, our work considers a variety of features, including demand, time-based, and weather features, at a group of public charging stations, while also studying network and individual station medium term demand.

Studies that predicted the demand in terms of specific EV users or used user behavior data include [15, 13, 19, 20, 10]. For instance, in [10], Luo et al. considered non-demand based features but only tested on a vehicle sharing platform for short-term demand forecasting. Their approach was therefore able to leverage features unique to individual drivers. Additionally, [15] used a statistical approach and found that the median energy consumption in a user's charging session history should be used to estimate demand. These studies generally assume the availability of data such as how long a user drove since their last charge, average kilometers driven per day, or user ID. Nonetheless, our use of anonymous data is less limiting due to concerns about data privacy, reconstruction of results, and applicability to public charging.

There is also considerable recent work in charging prediction which focuses on smart charging and scheduling of a fleet of EVs. The study by Mbuwir et al. [21] creates an optimized schedule for charging with reinforcement learning (RL). The study offers a novel control framework to minimize electricity cost, where a RL agent learns the control policy used to determine the real-time charging power of each EV. The study by Frendo et al. [14] predicts the charge profile of EVs with different learning methods. Their study predicts the short-term battery power draw as part of an algorithm to find an optimal charge schedule, with XGBoost achieving the best MAE (6%) for the charge profile predictions.

Some other various versions of the charging prediction problem include [22] which used clustering to identify different types of charging behavior and proposed an algorithm to quantify the amount of flexibility in charging at various times of day. [7] evaluates the impact of EV charging in a region on the distribution network in the near future, yielding categorical output known as a risk factor. A high risk factor was defined as implying a risk for the mid-term normal operation of the distribution network. Its contribution is a modeling framework and data analysis methodology but does not demonstrate the accuracy of the approach. The study by Ma and Faye [23] proposed a new LSTM based model to predict station occupancy in the immediate next time steps with high accuracy. The time steps used were either 10 minutes or 6 hours of public charging station occupancy in a city in the UK. [4] first predicted traffic flow, i.e. number of vehicles passing a point on a highway per hour, with a convolutional neural network (CNN), then predicted the demand at one fast charging station on the side of a highway with a queuing probabilistic model. A mean absolute percentage error (MAPE) of less than 25% was achieved for demand in the next day, but the study assumes the availability of traffic flow and personal data (e.g. distance traveled before the station, daily travel distance, etc) and only explores short-term predictions. Overall, the framework we present advances the state-of-the art by demonstrating feature engineering strategies with automatic dataset creation, exploring medium term demand prediction, and evaluating the performance of different features and models.

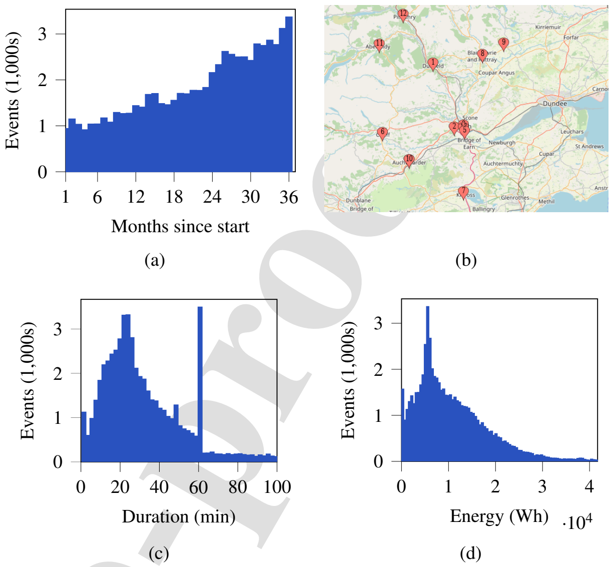

Figure 1: (a) Distribution of charging events in time since start of collection. (b) Locations of the 13 stations. (c) Charging event distribution of duration. (d) Charging event distribution of energy consumed. We trimmed ~1% of the events from plots (c) and (d) for visual purposes only because they were outliers. These outliers remain in the dataset.

3. Charging Events

Our approach considers charging data [5] with 66,664 charging events in a county in Scotland, UK. The county spans an area of 5,286 km^2. The data covers charging events over three years from September 1, 2016 to August 31, 2019. The distribution of the charging data over the three years is shown in Figure 1 (a), demonstrating a rapid year-over-year growth of EVs.

3.1. *Charging Stations*

There are 13 stations in the dataset. Two stations are dropped because they have little charging activity, having fewer than 400 charging events. The remaining eleven stations all have more than 1,000 events each. We define a station as a charging location with a unique longitude and latitude (+/- a margin of 25 meters). Each station has one or more charging outlets or charging station identifiers. The locations of the 13 stations are mapped in Figure 1 (b). The charging stations are not all in the same city, and the most secluded station is 50km from its nearest neighboring station. We assume these stations are all part of the same charging network.

3.2. *Charging Details*

Charging events are characterized by the starting time, charging duration in seconds, and total energy consumption in watt hours. The distribution of energy and duration is shown in Figure 1 (c) and (d). Most events consume about 13kWh and most people stay at the station for less than one hour. One anomaly in the dataset is that two percent of the events have zero or near-zero energy. Another is that about the same percentage of events have zero duration. The only other anomaly is that in Figure 1 the duration distribution is different from that of energy. This is because duration in this sense means how much time the vehicle was connected to the port, not how much time active charging occurred. To be precise, $E = Pt$, where E is the energy in Joules per second, P is the power in watts, and t is the time in seconds. Overall, the dataset anomalies do not influence the results because the metric our study focuses on predicting is the aggregate energy, at both individual stations and in the whole network.

4. Methodology

Our methodology takes input from real-world data, prepares the data for input to models, trains machine learning algorithms, and outputs forecasts of the daily energy consumption, up to a week in the future. Generally the real-world data from energy service providers is organized as transactions, and significant data processing effort is needed before prediction methods can be applied. Thus much of our methodology is dedicated to automatic dataset creation and feature engineering.

Our solution includes a ML prediction pipeline as follows. The charging data from Section 3 is cleaned and split into sets for training, validation and testing. Then, feature extraction and labeling turns the list of charging events into a time-series dataset. Normalization is subsequently performed to put features and labels on the same scale. Next, a prediction model (e.g. ML model) is trained on the data (excluding the test set). Finally, the inverse of the normalization is performed, giving the prediction of future demand.

4.1. Feature extraction and labeling

Our goal is to predict the total energy consumption over all stations in the network for each of the next seven days, starting from a time in the future selected by the user. In this section, we discuss the first steps needed to make a prediction model, such as a neural network or decision tree, handle time-series data. We form training examples for models by collecting all charging events that occurred within a certain time interval and computing the total energy consumed during those events. In our use case, training examples are generated for a series of overlapping time intervals, and labeled with the energy demand in that period. Additionally, the practitioner can append a variety of features which allows for rapid feature exploration. Other dataset parameters can be manipulated at the same time, such as the prediction granularity and the length of charging history to consider in each training example. We do not assign more weight to time periods with more charging because low activity times are just as important to predict as high activity times.

For example, consider Figure 2. Time intervals are chosen to end at t_i, $i \in 0, ..., 7$. Charging events ($\{\boldsymbol{A}, ..., \boldsymbol{I}\}$) are collected, to create input features, from before the end of the interval, up to a depth of history D_h, e.g., $\boldsymbol{E}$, $\boldsymbol{D}$, and $\boldsymbol{C}$, from t_4 to t_1. The events that follow are collected, to create output labels, from the end of the interval to the prediction horizon H, e.g., $\boldsymbol{F}$, $\boldsymbol{G}$ and $\boldsymbol{H}$, from t_4 to t_6. Note that events are included or excluded in history (features) or prediction horizon (labels) based on their start time alone.

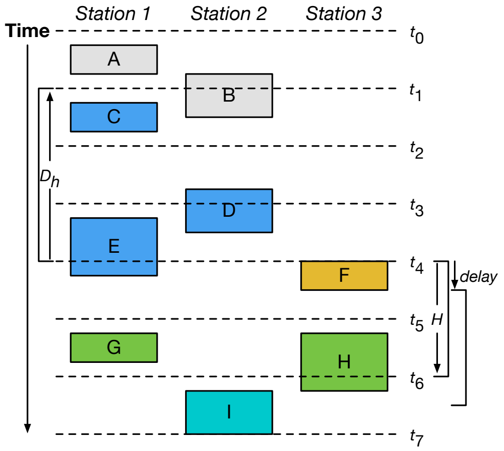

Figure 2: Charging events are grouped into overlapping intervals such that the features of past events (e.g., C, D, and E, in t_1 to t_4) can be used to predict future events (e.g., F, G, and H, in t_5 to t_6).

4.1.1. Prediction delay

We train models to predict demand for each day over the next seven days. However, a user may need to query the model for the demand at a time not exactly aligned with intervals we assume in the construction of training data. In this case, the model needs an additional input for the desired prediction delay.

Consider the following example. It's 9 AM, but we are interested in the demand for the next 24 hours starting at 12 PM. In this case, the first day of prediction starts three hours from now and ends 27 hours from now. We accommodate this shift in prediction horizon by adding new training examples for each time interval. These new examples have the same input features as the original (i.e., they use the same charging events to predict future behavior), except for a new feature, prediction delay. The output label is adjusted to account for the prediction delay by excluding charging events that no longer fall within the early part of the prediction horizon, and including charging events that now do fall within the late part of it. Looking once again at the example in Figure 2, a training example for the interval ending at t_4 with prediction delay uses E, D, and C to create input features, but G, H, and I for its output label: F is excluded, as it starts before $t_4 + delay$, and I is included, because it starts before $t_4 + delay + H$.

We identified several possible ways to generate new training examples with prediction delays:

1. For each time interval, a random prediction delay is selected from a continuous range of values, and one new training example is formed.

2. For each time interval, a random prediction delay is se-

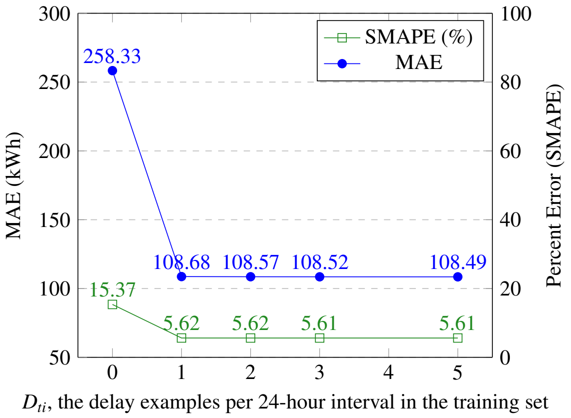

Figure 3: Prediction of energy consumption with a neural network on the validation set with and without prediction delay. The validation set is held constant at five delay samples per time interval.

lected from a discrete range of values, and one new training example is formed.

3. For each time interval, D_{ti} random prediction delays are sampled from a continuous range and D_{ti} corresponding training examples are formed.
4. For each time interval, all possible prediction delays in a discrete range are used.

We implemented Option 3 above because random prediction delay is expected to best represent user behavior, and multiple delayed samples per training examples was found to perform the best. Prediction delay is sampled from a uniform distribution ranging from 0 seconds to 24 hours, exclusive.

Figure 3 plots SMAPE and MAE for the demand in the entire network as a function of the number of prediction delay samples. The results are observed on the validation set, which spans five months. We observe that using more delay samples for each time interval improves accuracy; with just one delay sample per interval, an ANN is able to learn the offset. However, from the data we choose five samples for slightly improved accuracy.

4.1.2. Dataset parameters

The dimensions of the model and the datasets formed from the charging events are controlled by a number of parameters (Table 1).

Table 1: Parameters affecting dataset construction.

Parameter	Description
N_{ti}	Number of time intervals
D_{ti}	Number of delay samples per time interval
D_h	The depth of demand history to consider
D_{wp}	The depth of past weather information to consider
D_{wf}	The depth of future weather information to consider
H	The prediction horizon (e.g., in days)
F	Input feature set (e.g., Table 2)
L	Output label set (e.g., daily network energy demand)

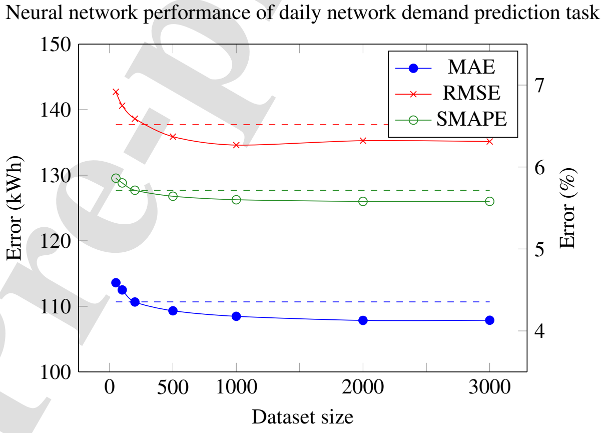

Figure 4: Neural network performance on the validation set as we vary the number of times the dataset is sampled, i.e., the number of time intervals N_{ti}. Baseline (average of past behavior) error is shown as dashed line.

The number of training examples affects how much labeled data the model can learn from. Figure 4 shows how the number of training examples affects the performance of our model, when the parameters of the validation set are held constant. The number of training examples is equal to the number of time intervals, N_{ti}, assuming no prediction delay is used. The dashed line shows each corresponding error curve when the average of the past week's charging demand is used to predict future daily demand. We observe that a N_{ti} of two thousand achieves the best performance across the error metrics. Since more does not help and less performs more poorly when looking at MAE and SMAPE error metrics, we will take two thousand as the value for the number of time intervals, resulting in a new interval about every 10 hours.

Similarly, we show how different depth of demand history, D_h, values affect the prediction of network energy performance in Figure 5. Here validation error is plotted with D_h varying from seven to thirty days. Prediction error tends to reduce slightly or remain constant as we look at greater depths of history. We choose a D_h value of 14 days because we observe that there is no performance benefit with more history, while using more history leads to more training time.

4.1.3. Feature selection

A summary of the features used is shown in Table 2. A natural choice of feature is past data. Since the target is daily energy consumed, we use past energy as a feature, listed in the first row of Table 2. Next we include some features that encode time, such as the time of day, time of year, or weekend vs weekday. Features such as these can represent travel periods such as holidays. Additionally there are a few more demand-based features such as event duration. Lastly, we have the past and future weather data.

Table 2: Features used in the models.

Feature	Description	Units
Energy; E_{Dh}, ..., E_{t-2}, E_{t-1}	Past aggregate energy	Wh
Month	Month of year	One-hot-encoded
Day of Week	Day of the week of the training example	One-hot-encoded
Time of Day	The time of day (10:00, 16:00,...) of the training example	Seconds
Calendar Day	Time elapsed since January 1, 1970	Seconds
Time of Year	Time elapsed since January 1 of the current year	Seconds
Prediction Delay	Time buffer between time of features and labels	Seconds
Weather	The local weather data for the time preceding or succeeding the current time	Various, see Table 3
Average Duration	The average duration of an event in a timeframe	Seconds
Average Energy	The average energy at a station or network in a timeframe	Wh
Time-averaged Energy	Average Duration * Average Energy	Wh^2

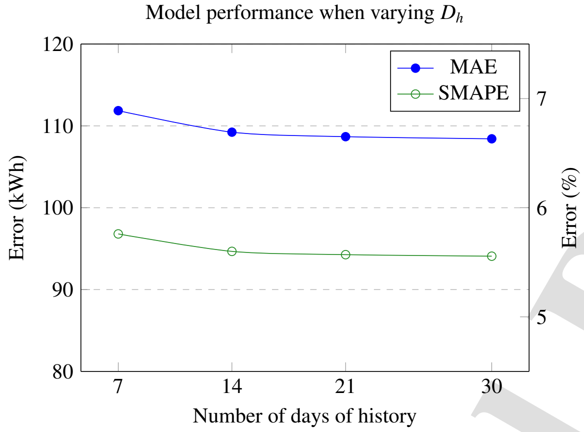

Figure 5: Validation error for daily network demand prediction with increasing depth of demand history D_h. More historical demand history can decrease prediction error.

Table 3: Description of weather variables used in models.

Variables	Description	Units
Max Temp	High temperature on a calendar day	°C
Average Temp	The average temperature recorded on a day	°C
Total Precip	The total precipitation, rain or snow, that fell on a calendar day	mm
Cloud Coverage	The percent of maximum cloud cover. In the range [0, 100] where 100 means there is no clear sky visible	%
Wind	The average wind on a calendar day	km/h

4.1.4. Weather

Two reasons weather could be important for charging prediction are first because it affects the performance of the EV, and second because it can stimulate travel. For instance, the range of EVs is less as temperature falls, while charging time increases. Furthermore, [7] established a statistical relationship between weather and charging station demand and hypothesized that in a cold climate such as the UK's, energy requirements increase due to the need to heat the vehicle cabin.

We used weather data from a central location in the region from an online database [24]. Since the geographical area is not large and the number of weather stations is limited, all stations are assumed to have the same weather. The weather data used for a forecast is shown in Table 3. Our study compares the use of models with weather vs no weather, where with weather means using all the features from Table 3.

Here we consider variables that may affect the performance of the vehicle, including temperature. Another component that affects vehicle performance is the use of air conditioning, so variables such as cloud cover are included to account for the fact that the greenhouse effect makes it hotter in the cabin. In general, we use a simple strategy of choosing variables that plausibly effect charging, checking assumptions by looking at model performance, and using regularization techniques to reduce overfitting.

Note that our experiments assume accurate weather predictions because the weather data used here are ground truth values, not the weather forecasts that would be available to the model at test time. No historical weather forecast data was available for the times and locations of the charging stations used in this study. While we have made the simplifying assumption that there is no discrepancy between the actual weather and weather forecasts, in the future we hope to study the impact of such discrepancies on demand prediction accuracy.

Figure 6 shows that model performance increases as the number of days of future weather data, D_{wf}, approaches three days. With more than three days of data the performance appears constant or worsens possibly due to overfitting. We choose to keep this parameter at three days without the use of a noise model.

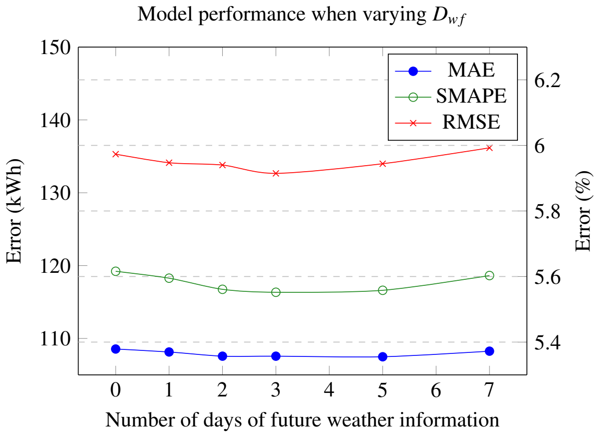

Figure 6: Validation error of network prediction task with different weather horizons.

4.1.5. Normalization

As a last step, features are put on the same scale using

$$z_{norm} = \frac{z - min(z)}{max(z) - min(z)} \tag{1}$$

to eliminate bias towards a particular variable.

Here we normalize depending on the features found in the training set (2016 - 2018 data). Note that in general the electrical load of a system increases continually [8]. Thus because charging activity can increase, data is not guaranteed to be on the scale [0, 1]. The implication of this is that model behavior could change to give more weight to larger values if typical energy values increase significantly in the next months or years. Even though this situation is foreseen with the expected growth in charging, it is easily overcome with periodic retraining.

4.2. Multi-task learning

In our approach, we use multi-task learning (MTL) to predict multiple outputs simultaneously. MTL is an inductive transfer mechanism that aims to improve generalization performance. MTL has been shown to be beneficial to neural networks in general, with the main benefit that overfitting can be reduced, and thus accuracy can be improved, compared to single-task learning [25].

Multi-task learning arises when a situation requires obtaining multiple predictions at once. For example, in drug discovery, tens or hundreds of active compounds should be predicted. In that domain, [26] found that MTL monotonically increases the accuracy as the number of tasks increases. Even when one only cares about the performance of a single task, the use of auxiliary tasks can increase performance. Commonly, the model learns tasks that are closely related to the main task, allowing the model to learn a beneficial representation [27].

We use a single neural network for a few strongly related tasks. This approach is not new, and has been demonstrated by Sejnowski and Rosenberg in their application to learn both phonemes and their stresses [28, 25]. Our methodology uses MTL to leverage information about charging demand in one location to improve the prediction at another.

We use the algorithm presented in Algorithm 1 to iterate over the different stations during feature extraction, to capture correlated demand at different stations in the network. The demand at each station is a different task. The last station visited in this algorithm is a dummy station representing the task of predicting network demand. The features and target of this station include the sum of the demand at all stations. The model parameters are shared across all tasks, i.e., we employ hard parameter sharing.

Algorithm 1: Form labeled dataset

```
Input: Dataset of charging events with basic charging
       data for each event
Output: Labeled dataset for a charging network with
        $N_S$ many stations
Initialize: Row index of new, labeled dataset: $i$
  Time-series parameters: $D_h$, $D_{wp}$, $D_{wf}$, $D_{ti}$, $N_{ti}$ with list
  of time intervals $list_{N_{ti}}$
for $t$ in $list_{N_{ti}}$ do
    Initialize a list of random delay values, $list_{Dti}$, of size
      $D_{ti}$
    for $d$ in $list_{D_{ti}}$ do
        $dataset[i][featureIndex] \leftarrow d$
        $dataset[i][featureIndices] \leftarrow$ non-station
          specific features, labels corresponding to time
          $t + d$
        for $s$=1 to $N_s$ do
            $dataset[i][featureIndices \cup$
              $stationIndices] \leftarrow$ station specific features,
              labels corresponding to time $t + d$
        $i \leftarrow i + 1$
```

First, parameters are initialized and a list of time intervals is generated. Then, we iterate through the list of time intervals, followed by a list of delay values. Next all the non station-specific features are extracted for the given time and appended to the row. Then, we iterate through the stations and append the station-specific features to the training example. We show later how stations can be modeled individually, making the last step optional. The output of Algorithm 1 for each training example can then be written in matrix representation as $example_{mat}$ =

$$\left(\begin{array}{cccc|ccc} x_{11} & x_{12} & \cdots\cdots & x_{1N_x} & y_{11} & \cdots & y_{1N_y} \\ x_{21} & x_{22} & \cdots\cdots & x_{2N_x} & y_{21} & \cdots & y_{2N_y} \\ \vdots & \ddots & & & & & \\ x_{N_s1} & x_{N_s2} & \cdots\cdots & x_{N_sN_x} & y_{N_s1} & \cdots & y_{N_sN_y} \end{array}\right)$$

where N_x and N_y are the number of columns in the labeled dataset corresponding to the features and labels respectively. More precisely, we have $N_x = \sum_{i=1}^{|F|} D_{h_i} * |f_i|$ and $N_y = \sum_{i=1}^{|L|} D_{h_i} * |l_i|$, where $f_i \in F, l_i \in L$. Then, *example* is finally formed by flattening $example_{mat}$ to make the data 1-dimensional, allowing for seamless input to most prediction algorithms, e.g., ANN or random forest. The number of rows in the dataset is

$$|dataset|_r = N_{ti} * (D_{ti} + 1) \quad (2)$$

and the number of columns in the dataset is

$$|dataset|_c = (N_x + N_y) * N_s \quad (3)$$

Besides potentially higher accuracy, a second natural benefit of our MTL implementation is generating additional outputs at the same time. Moreover, we generate the demand at multiple stations on multiple days at the same time. This is beneficial because the alternative is to tune, train, and maintain many ML models. We do not anticipate such an approach would scale well, particularly with the expected rapid growth in EV charging. For instance, industry figures show the EU had about 220,000 public stations in 2020. However, the region's automotive and sustainable transport lobbies told the EU to expect 1 million by 2024, and 3 million by 2029 [29].

Finally, this means in our approach the demand in different areas is obtained with just one inference operation. Also, simply summing the outputs will obtain the demand at any combinations of stations, as they are obtained directly from the right-hand side of $example_{mat}$. Thus overall we provide a framework for stations focusing attention on other station's behavior, improving prediction for stations or networks, and generating output for multiple tasks.

5. Experimental Setup

We performed a series of experiments to evaluate the ability of different machine learning approaches to forecast daily charging station energy consumption. Our experiments evaluate several different models, the utility of weather features, and different options for the output layer. First, we compared the performance of a few machine learning models, such as neural networks and random forests, to predict the aggregate network energy. Meanwhile, we studied multiple output layer structures, because the output layer of a neural network can be common to each individual station, or shared amongst stations. Then, we compared those results with that of the multitask learning strategy, and also measured the effects of using weather features. Finally, we performed the above experiments a second time, but assumed the user is concerned with the aggregate energy at individual charging stations, rather than the energy in the network.

5.1. Testing Strategy

To test the effectiveness of the approach the dataset is divided into training, validation, and test sets. The division into these sets occurs for example by assigning each interval in Figure 2 to a set, where the sets correspond to one of three time periods in Figure 7. As per Figure 7, the training set contains 66.1% of the charging events, and covers 77.8% of the time period spanned by the dataset, while the validation and test set contain smaller percentages. We choose a testing strategy suitable for time-series applications known as last block validation [11].

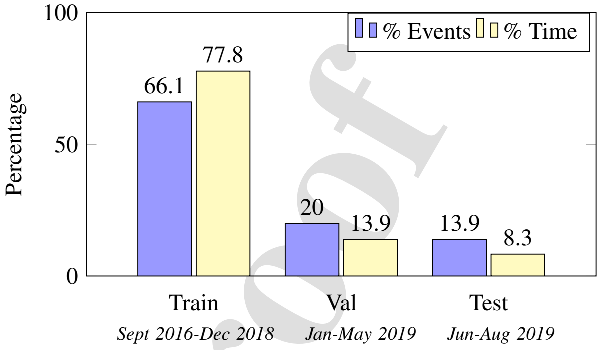

Figure 7: Data split distribution. Data is split into separate time periods for training, validation, and testing. The training set covers all seasons, while we validate and test on winter through summer.

The strategy ensures validation and testing occur on a mix of seasons of the year (winter through summer), as depicted in Figure 7; we plan to also validate on fall data when more data becomes available. The testing strategy enables us to find a robust model where separate seasons do not need their own prediction models. Note that the model does not have access to any data from the period of time from which test set samples are selected.

5.2. Approaches

We trained the following set of machine learning models on the dataset, with hyperparameters tuned on the validation set:

- artificial neural networks (ANN) [30];
- random forests [31], with number of estimators and maximum depth as hyperparameters;
- k-nearest neighbors (kNN); [32], with hyperparameter k;
- linear regression (LR);
- support vector regression (SVM or SVR) [33], with degree as a hyperparameter and also the kernels as radial basis function, linear, or polynomial.

The full feature set is used to train all machine learning models.

We also use ARIMA [34], a classic time-series forecasting model, in our experiments and choose the following baselines to determine if any useful learning takes place:

1. Baseline 1 - The data from the previous week is the forecast for next week, i.e., we have $\{E_{t+0}, E_{t+1}, ..., E_{t+6}\} \leftarrow \{E_{t-7}, E_{t-6}, ..., E_{t-1}\}$.
2. Baseline 2 - The average of the previous week's data is forecast as the demand for all of the next week, i.e., we have $\{E_{t+0}, E_{t+1}, ..., E_{t+6}\} \leftarrow avg\{E_{Dh}, E_{Dh-1}, ..., E_{t-2}, E_{t-1}\}$.

where E_{t+0} is the first day of prediction.

5.3. Loss Functions

We define the loss function for training as the mean squared error:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2, \quad (4)$$

with equal weight given to each output. Many loss functions can be defined to report results for regression tasks. The loss functions we include here are the mean absolute error (MAE), root mean squared error (RMSE) and symmetric mean absolute percentage error (SMAPE). MAE and RMSE are universal and can be applied to any regression task, for instance energy consumption, charging duration, etc. We include SMAPE because it is a common percentage metric and as such considers the error as a percentage of the target. These three metrics are defined as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|Y_i - \hat{Y}_i| \quad (5)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2}{n}} \quad (6)$$

$$SMAPE = \frac{100\%}{n} * \sum_{i=1}^{n}\frac{|\hat{Y}_i - Y_i|}{|\hat{Y}_i| + |Y_i|} \quad (7)$$

5.4. Network vs Station Level Modelling

Again, our study includes two ways of applying machine learning to predict charging demand. Firstly, we explore network demand, which is the sum of the demand at all stations in the network. Network demand can be modelled in multiple ways. Table 4 describes three ways to model network demand we studied (see the first three rows), with each row also indicating if the strategy uses MTL. Method A is the most simple and does not attempt to use MTL. In contrast, Methods B and C do leverage MTL and organize the tasks to form a dataset exactly as in Algorithm 1.

In addition to considering network demand, we also deploy our methodology to model an individual station's demand. In terms of output layer structure, stations can be either modelled individually (Method D), or modelled with other stations or tasks (Method E), where the output layer takes a matrix format. For this matrix, the number of rows and columns is given by equations (2) and (3), respectively. When deciding how to implement Method D, we decided that this method should use a single neural network model applied to all stations, because this strategy is easiest to scale compared to the charging station at each location getting its own model that has to be trained and fine-tuned.

6. Results

We conducted a series of experiments to evaluate different approaches to modeling the prediction of public charging demand. We include the results of experimenting with different prediction models, features, and output layer structures. We divide the results into two sections, first for the prediction in the whole network and second at individual stations. Lastly, we include a discussion to interpret the results.

Table 4: Ways to model demand with varying output layer structure.

Method	Network/ Station	MTL	Description
A	Network	No	Sum demand at all stations and input to a model dedicated to this task.
B	Network	Yes	Input each station's demand to a model individually, in addition to the network demand, and take the dedicated network demand label as the output, ignoring the station output tasks.
C	Network	Yes	Same as B, but instead of taking the dedicated network label, ignore it and assume the network demand is the sum of individual station's demand.
D	Station	No	Each station gets its own model. Prediction of demand at that station is its own task.
E	Station	Yes	Each station is part of a shared model, as in Method B and C.

6.1. Performance

6.1.1. Network Demand

We present the results of predicting the charging demand in the network in Table 5 and Table 6. Table 5 shows the results when forecasting network demand with single-task learning (Method A). With a single task, any of the machine learning models discussed can be applied without modification. The ANN performs the best, with about 7% percent better MAE and SMAPE than the best baseline (Baseline 2, which is the average of last weeks data). In Table 6 we show the performance on the task of predicting network demand with MTL. Recall from the description of the output layers (Table 4) that this approach is further divided into Method B and Method C. The ANN performs the best with Method C, with a 2% improvement over Method B. Moreover, the use of MTL for learning network demand led to a 1-2% improvement over Method A. The use of weather input led to a 7% performance improvement.

Finding the ANN hyperparameters used for the results in this study was guided by a grid search. The optimal ANN was a large net with three hidden layers with 2048, 1024, followed by 512 nodes, as well as two dropout layers. We also had the following hyperparameters: learning rate: 0.001; loss function: Mean Squared Error; Optimizer: Stochastic Gradient Descent.

6.1.2. Individual Station Demand

We present the results for predicting demand at individual stations in Table 7 and Table 8. Because there are multiple stations, here we assume the goal is the reduce the average error across all of them. There is no need to weight busier stations because SMAPE is a percentage error so it already provides relative error values.

Table 5: Network Demand Prediction using Method A

Model	MAE	RMSE	SMAPE
ANN	**125.481**	**158.192**	**5.964**
Linear Regression	127.431	160.031	6.040
Baseline 2	135.574	167.847	6.385
ARIMA	145.450	174.362	6.890
Random Forest	150.275	187.819	7.091
SVM	162.193	201.606	7.436
Baseline 1	167.595	209.761	7.929
KNN	370.265	408.069	20.551

Table 6: Predicting network demand for next week using Method B and C

	With weather		
Model	MAE	RMSE	SMAPE
ANN (B)	126.731	157.908	6.003
ANN (C)	**124.670**	**155.518**	**5.925**
	Without weather		
Model	MAE	RMSE	SMAPE
ANN (B)	135.068	168.035	6.300
ANN (C)	134.016	165.742	6.258

Single-task results for individual station prediction are presented in Table 7. The top performing models are the ANN and ARIMA. The ANN performs similarly to the ARIMA model in terms of MAE and RMSE. However, when considering SMAPE, the ANN performs much better (12% relative improvement or 3% absolute improvement). Moreover, the results show that weather data accounted for a 2% improvement in MAE for the best performing model, an ANN. Multi-task learning can predict the demand at every station and the network at the same time. We show the results for individual station demand prediction in Table 8. However Method E is slightly worse (by about 3%) than D across the three metrics. This shows that, overall across all stations, demand at each station was not useful as an auxiliary task when predicting individual station demand.

Table 7: Model comparison for Method D.

Model	Avg MAE	Avg RMSE	Avg SMAPE
ANN	**31.315**	39.649	**21.308**
ANN-*no weather*	31.888	40.308	22.116
ARIMA	31.979	**38.486**	25.247
KNN	57.117	46.579	33.350
Linear Regression	33.105	40.977	35.977
Random Forest	36.316	45.267	28.942
SVM	49.473	59.722	33.643
Baseline 1	42.593	53.661	33.629
Baseline 2	32.691	40.773	25.466

Table 8: Impact of weather features on individual station demand prediction using ANN with Method E.

	MAE	RMSE	SMAPE
Without weather	33.566	42.239	22.576
With weather	**32.591**	**41.088**	**22.351**

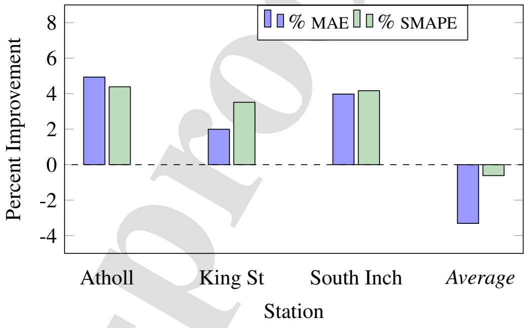

Figure 8: Stations with improvement using MTL (percent change from Method D to Method E).

We observe that multi-task learning was not beneficial for the station prediction task, but it was for the network prediction task. Interestingly, however, some station's forecasting accuracy can be improved with MTL Method E. Three out of eleven stations showed better results with Method E compared to D. This improvement was from two to five percent, as shown in Figure 8. In other words, this MTL technique, when applied to individual stations, was effective at the stations shown in Figure 8 and not the others. These three stations are not adjacent, and are rather located in different cities. A possible reason for the above behavior is that these three stations have similar popularity, with each having between 3,500 and 6,000 charging events total over three years. Studying additional reasons for the above MTL behavior seen would be interesting for future work.

Since the ANN with Method D performed the best at predicting individual station demand, we show the breakdown of each station's performance in Table 9. This table shows the variation in performance among the stations, where the error of most stations is around 20% SMAPE. However, the table shows that there was variation depending on the station. For example, the error at the Friarton station was higher at about 44% SMAPE.

Also, Figure 9 shows an example of predicted and actual values in the test set. The values are plotted next to Baseline 2, which is a moving historical average. For about half of the time period, the baseline performs the same or better than the model. Nonetheless, considering the magnitude of the error, as we have throughout the results, the overall accuracy is best with the ANN model.

To summarize, the lowest MAE from predicting network energy for seven days was 124.670kWh (Table 6), while in the test set the average daily energy consumed in the network was 1.086 MWh: the MAE while predicting the next seven days of demand is less than 12% of the average daily network de-

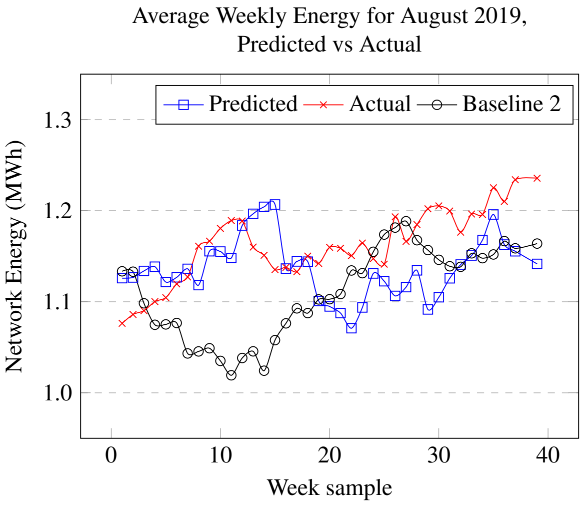

Figure 9: Model performance showing predicted vs actual and baseline 2 values for August 2019 in the test set.

Table 9: Breakdown of Station Results Test Error using ANN with Method D sorted by ascending MAE.

Station	MAE (kWh)	RMSE (kWh)	SMAPE (%)
Canal	12.530	15.846	30.644
Friarton	13.992	18.722	43.549
Leslie Street	20.891	27.820	25.331
Crown Inn	24.619	29.991	11.866
Mill	26.211	33.095	19.934
King Street	28.177	34.578	19.988
South Inch	32.685	40.364	20.333
Atholl	36.078	45.522	18.333
Rie-Achan	42.397	56.845	20.416
Kinross	52.884	66.158	11.866
Broxden	54.005	67.199	12.130
Average	*31.315*	*39.649*	*21.308*

mand. For comparison, the study in [4] used a novel probabilistic queuing model to predict charging load with a percentage error of 3-4%, but used a prediction horizon of 1 hour. For next day predictions, the MAPE was about 25%. This study also relied on traffic information, which may not be as readily available as weather information.

The lowest average error for predicting individual station demand was achieved by an ANN with Method D, with a MAE of 31.315 kWh. This was only about four percent less error than Baseline 2. However, the results here are across all stations in the network, and the standard deviation in the MAE is $\sigma = 13.3$kWh, which means the ANN method benefits some stations more than others. One possible reason for this is that while each station has its own model, it did not get its own hyperparameters specifically tuned for it, because such a design would not scale well. At the same time, the stations have different sparsity levels and demand profiles.

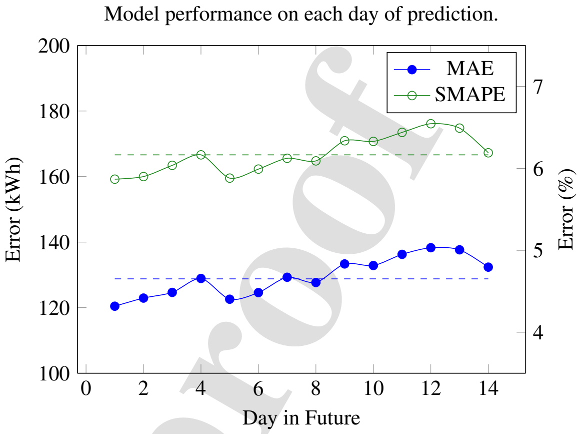

Figure 10: Model performance of network prediction task with expanded prediction horizon. The dashed line is a baseline which is the average of past behavior.

6.2. *Discussion*

We showed that, in the application area of public car charging, machine learning can train on historical data, combined with various features, to outperform ARIMA, a classic time-series forecasting model. We showed that machine learning can perform accurate forecasts, with an ANN showing the highest potential in each application area when compared to the other models. Notably, when we choose features to form a dataset which models then train on, we outperform the ARIMA model and simple baselines which inherently do not exploit features.

Additionally, we consistently saw that weather data improves predictions. In our best performing model/method, the performance improvement was 7% when the target was network demand, and 2% for individual station demand. Furthermore, because none of our baselines can represent non-demand based features, in a geographical region where weather is a greater factor in determining charging demand, the performance gap between statistical models and machine learning could widen. We took a first step towards using weather by incorporating a few days of past and future weather data. In future work, one could use larger values of D_{wf} with data augmentation to leverage more data. Such a strategy would also consider that weather forecasts can be very inaccurate more than a few days ahead. For future study, a successful implementation might require an accurate noise model to perturb the weather data, since forecasts drop in accuracy significantly with prediction horizons longer than a few days.

We also investigated extended prediction horizons. We adapted the model to predict 14 days ahead. Figure 10 shows the performance on each day with the extended horizon, with the baseline being the average of the past 14 days shown as a dashed line. The overall trend shows more accurate predictions closer in the future ($R^2 = 0.79$), while also showing that the model performs better or equivalent to baseline up to eight days in the future.

Furthermore, the results show that a one model per station strategy can be used, returning a reliable forecast, like has been demonstrated before for short term predictions [11]. We saw that a one model per station strategy, compared to a shared model, works better on average, as shown by comparing Table 7 and Table 8.

7. Conclusion

In this study we demonstrated the effectiveness of machine learning to make predictions of future public EV station demand for up to a week in advance. We formulated the prediction of charging activity as a regression problem, and quantified demand as the daily energy consumption. In our solution, we proposed a time-series inspired method to generate labeled data from a set of charging events. We presented a framework that automates dataset creation and labeling. This framework also allowed us to explore output layer structures and find useful features for predicting the medium-term public charging demand. We observed that increasing the amount of historical data considered, up to two weeks, improved results.

In the presentation of our framework, we presented novel feature engineering algorithms for medium-term charging demand prediction. Specifically, we observed that including additional training examples to support flexible (delayed) prediction horizons, improved model performance: users may query the model for arbitrary times of interest, and expanding the data set accommodated this. Furthermore, later, we demonstrated that multi-task learning strategies applied to the charging dataset can also improve prediction performance.

Comparing the accuracy of our model to a few simple baselines showed promising results using various performance metrics, namely MAE, RMSE, and SMAPE. Our ANN model outperformed statistical methods, including ARIMA, simple baselines, and other machine learning methods. Compared to simple baselines, our ANN model achieves 7% lower MAE when making network demand predictions. Comparing amongst the machine learning and statistical approaches when predicting network demand, our ANN outperformed the next best model, linear regression, by 1.5%. When predicting individual station demand, our approach outperformed ARIMA by 2%. Moreover, we observed that using features based on weather data improved the network demand prediction by about 3%. Lastly, we transformed the dataset with our labeling strategy to investigate a form of multi-task learning, where different stations represent multiple tasks. While this was not beneficial predicting at individual stations, it improved prediction performance by about 1% for the network charging demand.

Acknowledgments

We would like to express our appreciation for the Natural Sciences and Engineering Research Council of Canada (NSERC) for supporting this work though grant number EGP 543736-19 and the Engage program.

References

[1] "U.K. to ban new gas cars by 2030 as leader touts green industrial revolution," *The Associated Press*, November 2020.

[2] D. Jones, "General motors aims to be carbon neutral by 2040," *NPR*, January 2021.

[3] M. G. Flammini, G. Prettico, A. Julea, G. Fulli, A. Mazza, and G. Chicco, "Statistical characterisation of the real transaction data gathered from electric vehicle charging stations," *Electric Power Systems Research*, vol. 166, pp. 136–150, 2019.

[4] X. Zhang, K. W. Chan, H. Li, H. Wang, J. Qiu, and G. Wang, "Deep-learning-based probabilistic forecasting of electric vehicle charging load with a novel queuing model," *IEEE Transactions on Cybernetics*, pp. 1–14, 2020.

[5] Open-Data Team, "Electric vehicle charging station usage." *Open Data Perth and Kinross*. Accessed July 10, 2020. [Online], 2017.

[6] S. Shahriar, A. R. Al-Ali, A. H. Osman, S. Dhou, and M. Nijim, "Prediction of ev charging behavior using machine learning," *IEEE Access*, vol. 9, pp. 111576–111586, 2021.

[7] E. Xydas, C. Marmaras, L. Cipcigan, N. Jenkins, S. Carroll, and M. Barker, "A data-driven approach for characterising the charging demand of electric vehicles: A UK case study," *Applied Energy*, vol. 162, pp. 763–771, 01 2016.

[8] H. Alfares and N. Mohammad, "Electric load forecasting: Literature survey and classification of methods," *International Journal of Systems Science - IJSySc*, vol. 33, pp. 23–34, 01 2002.

[9] G. Gross and F. Galiana, "Short-term load forecasting," *Proceedings of the IEEE*, vol. 75, no. 12, pp. 1558–1573, 1987.

[10] M. Luo, B. Du, K. Klemmer, H. Zhu, H. Ferhatosmanoglu, and H. Wen, "D3P: Data-driven demand prediction for fast expanding electric vehicle sharing systems," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 4, Mar. 2020.

[11] M. Majidpour, C. Qiu, P. Chu, R. Gadh, and H. R. Pota, "Fast prediction for sparse time series: Demand forecast of EV charging stations for cell phone applications," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 1, pp. 242–250, 2015.

[12] S. Shahriar, A. R. Al-Ali, A. H. Osman, S. Dhou, and M. Nijim, "Machine learning approaches for EV charging behavior: A review," *IEEE Access*, vol. 8, pp. 168980–168993, 2020.

[13] Y.-W. Chung, B. Khaki, T. Li, C. Chu, and R. Gadh, "Ensemble machine learning-based algorithm for electric vehicle user behavior prediction," *Applied Energy*, vol. 254, p. 113732, 2019.

[14] O. Frendo, J. Graf, N. Gaertner, and H. Stuckenschmidt, "Data-driven smart charging for heterogeneous electric vehicle fleets," *Energy and AI*, vol. 1, p. 100007, 2020.

[15] Tianyang Zhang, Xiangyu Wang, Chi-Cheng Chu, and R. Gadh, "User demand prediction and cloud-based smart mobile interface for electric vehicle charging," in *2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, pp. 348–352, 2016.

[16] M. Majidpour, C. Qiu, P. Chu, R. Gadh, and H. R. Pota, "Modified pattern sequence-based forecasting for electric vehicle charging stations," in *2014 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, pp. 710–715, 2014.

[17] X. Huang, D. Wu, and B. Boulet, "Ensemble learning for charging load forecasting of electric vehicle charging stations," in *2020 IEEE Electric Power and Energy Conference (EPEC)*, pp. 1–5, IEEE, Nov 2020.

[18] Z. Ye, R. Wei, and N. Yu, "Short-term forecasting for utilization rates of electric vehicle charging stations," in *2021 IEEE International Smart Cities Conference (ISC2)*, pp. 1–7, 2021.

[19] Y. Zhao, Z. Wang, Z.-J. M. Shen, and F. Sun, "Data-driven framework for large-scale prediction of charging energy in electric vehicles," *Applied Energy*, vol. 282, p. 116175, 2021.

[20] Y. Liu, W. Liu, S. Gao, Y. Wang, and Q. Shi, "Fast charging demand forecasting based on the intelligent sensing system of dynamic vehicle under evs-traffic-distribution coupling," *Energy Reports*, vol. 8, pp. 1218–1226, 2022. ICPE 2021 - The 2nd International Conference on Power Engineering.

[21] B. V. Mbuwir, L. Vanmunster, K. Thoelen, and G. Deconinck, "A hybrid policy gradient and rule-based control framework for electric vehicle charging," *Energy and AI*, vol. 4, p. 100059, 2021.

[22] N. Sadeghianpourhamami, N. Refa, M. Strobbe, and C. Develder, "Quantitive analysis of electric vehicle flexibility: A data-driven approach," *In-*

ternational Journal of Electrical Power Energy Systems, vol. 95, pp. 451–462, 2018.

[23] T.-Y. Ma and S. Faye, “Multistep electric vehicle charging station occupancy prediction using hybrid lstm neural networks,” *Energy*, vol. 244, p. 123217, 2022.

[24] “World weather online,” *WorldWeatherOnline.com*, 2021.

[25] R. Caruana, “Multitask learning,” *Machine Learning*, vol. 28, 07 1997.

[26] B. Ramsundar, S. Kearnes, P. Riley, D. Webster, D. Konerding, and V. Pande, “Massively multitask networks for drug discovery,” *arXiv:1502.02072*, 02 2015.

[27] S. Ruder, “An overview of multi-task learning in deep neural networks,” *CoRR*, vol. abs/1706.05098, 2017.

[28] T. J. Sejnowski and C. R. Rosenberg, “Nettalk: A parallel network that learns to read aloud,” Tech. Rep. JHU/EECS-86/1, John Hopkins University Department of Electrical Engineering and Computer Science, 1986.

[29] V. Eckert, “EU told 1 million public EV charging stations needed by 2024,” *Reuters*, February 2021.

[30] W. S. McCulloch and W. Pitts, “A logical calculus of the ideas immanent in nervous activity,” *The bulletin of mathematical biophysics*, vol. 5, no. 4, pp. 115–133, 1943.

[31] L. B. Statistics and L. Breiman, “Random forests,” in *Machine Learning*, pp. 5–32, 2001.

[32] N. S. Altman, “An introduction to kernel and nearest-neighbor nonparametric regression,” *The American Statistician*, vol. 46, no. 3, pp. 175–185, 1992.

[33] H. Drucker, C. Burges, L. Kaufman, A. Smola, and V. Vapnik, “Support vector regression machines,” *Adv Neural Inform Process Syst*, vol. 28, pp. 779–784, 01 1997.

[34] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

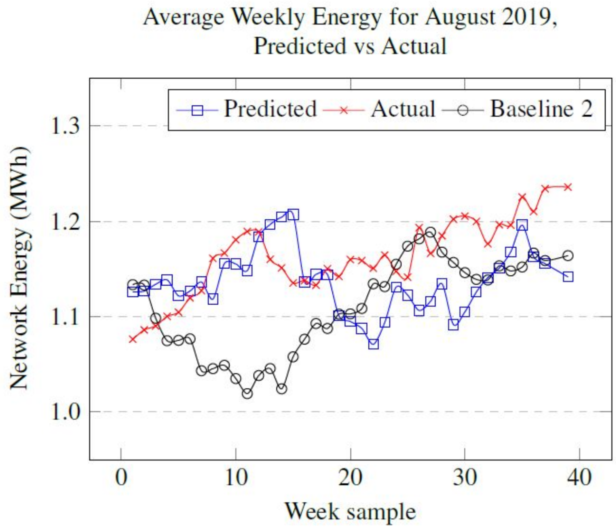

Figure 9: Model performance showing predicted vs actual and baseline 2 values for August 2019 in the test set.

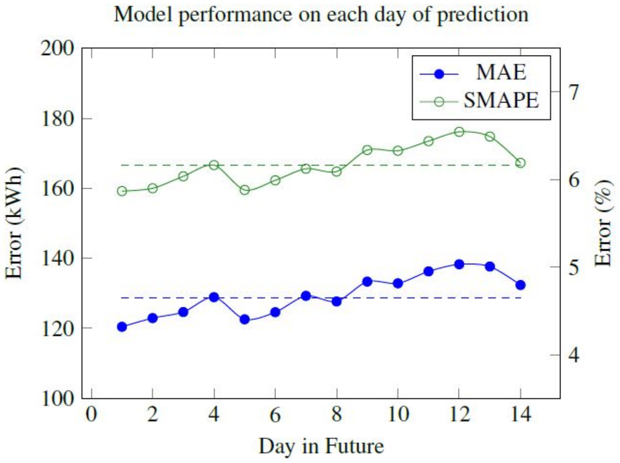

Figure 10: Model performance of network prediction task with expanded prediction horizon. The dashed line is a baseline which is the average of past behavior.

Declaration of interests

☒ The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐ The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: