

“© 2019 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.”

# Energy Demand Prediction with Federated Learning for Electric Vehicle Networks

Yuris Mulya Saputra<sup>1</sup>, Dinh Thai Hoang<sup>1</sup>, Diep N. Nguyen<sup>1</sup>, Eryk Dutkiewicz<sup>1</sup>,  
Markus Dominik Mueck<sup>2</sup>, and Srikathyayani Srikanteswara<sup>2</sup>

<sup>1</sup> School of Electrical and Data Engineering, University of Technology Sydney (UTS), Australia

<sup>2</sup> Intel Corporation

**Abstract**—In this paper, we propose novel approaches using state-of-the-art machine learning techniques, aiming at predicting energy demand for electric vehicle (EV) networks. These methods can learn and find the correlation of complex hidden features to improve the prediction accuracy. First, we propose an energy demand learning (EDL)-based prediction solution in which a charging station provider (CSP) gathers information from all charging stations (CSs) and then performs the EDL algorithm to predict the energy demand for the considered area. However, this approach requires frequent data sharing between the CSs and the CSP, thereby driving communication overhead and privacy issues for the EVs and CSs. To address this problem, we propose a federated energy demand learning (FEDL) approach which allows the CSs sharing their information without revealing real datasets. Specifically, the CSs only need to send their trained models to the CSP for processing. In this case, we can significantly reduce the communication overhead and effectively protect data privacy for the EV users. To further improve the effectiveness of the FEDL, we then introduce a novel clustering-based EDL approach for EV networks by grouping the CSs into clusters before applying the EDL algorithms. Through experimental results, we show that our proposed approaches can improve the accuracy of energy demand prediction up to 24.63% and decrease communication overhead by 83.4% compared with other baseline machine learning algorithms.

*Keywords*- Energy demand, electric vehicle, charging station, federated learning, clustering.

## I. INTRODUCTION

The electric vehicles (EVs) have emerged as one of the sustainable solutions to transform the conventional transportation systems with less oil use, high energy efficiency, and low gas emissions. According to International Energy Agency [1], the number of EVs on the road will rise tremendously by more than 4000% in 2030. This trend will lead to an explosion of energy demand in the power market and make a significant impact on the power grid system.

Generally, the power grid supplies the energy for charging stations (CSs) once receiving requests from EVs [2]. However, this approach experiences a serious energy transfer congestion when a huge number of EVs charging the energy simultaneously [3] and lead to high energy transfer cost for the charging station provider (CSP). To deal with this problem, energy can be reserved in advance at the CSs to meet real-time demands from the EVs [4]. Particularly, in [4], the authors proposed a solution to orchestrate EVs' charging using offline scheduling scheme. However, without considering the real demand history, this approach may suffer from under/over-utilization, due to the unpredictable and dynamic EVs' energy demands.

Hence, effective and intelligent approaches to predict energy demands for CSs in the EV networks are of significance to optimize energy efficiency. As such, the CSP can reduce the energy transfer cost and stabilize the energy demand in the power grid system. Furthermore, the CSP can provide real-time energy charging and guarantee stable energy supply for the upcoming EVs.

To predict energy demand for EV networks, the authors in [5] introduced a mobile application which can estimate energy availability at each CS using k-nearest neighbor (kNN) algorithm. Specifically, the history of consumed energy and charging duration for connected EVs is captured as the dataset and then stored on the remote server. In [6], the authors proposed a reinforcement learning-based demand response scheme to optimize the amount of energy charging for an individual EV based on daily forecasted price policy. Furthermore, the authors in [7] developed a driving activity-based recommendation system applying a multiple regression-based learning approach, aiming at improving the energy consumption prediction accuracy for EVs. In [8], an online learning to optimize EVs' charging demands using previous-day pricing profiles from the distribution company was proposed. Alternatively, the authors in [9] designed a smart charging policy for EVs using machine learning tools including deep neural network (DNN), shallow neural network (SNN), and kNN to decide the charging time when the EVs are connected to the CSs. Nonetheless, these approaches only consider energy demand prediction independently at each EV or CS, and thus they may not be effective for the whole EV network. In fact, predicting the energy demand using shared information or global models can further improve the prediction accuracy. Furthermore, those machine learning-based solutions do not address the communication overhead and privacy issues, which are very crucial in the future EV networks.

In this paper, we introduce state-of-the-art machine learning-based approaches which can not only significantly improve the accuracy of energy demand prediction, but also remarkably reduce the communication overhead for EV networks. In particular, we first introduce a communication model using the CSP as a centralized node to gather all information from the CSs in a considered area. We then develop an energy demand learning (EDL)-based solution utilizing deep learning method to help the CSP accurately predict energy demands for the CSs in this area. However, this approach requires the CSs to share their local data with the CSP, and thus it may suffer from serious overhead and privacy issues. To



for epoch time  $\phi$  (i.e., the time when all transactions of  $\mathbf{X}_{csp}$  has been observed) in the CSP is expressed as follows:

$$\rho(\mathbf{v}^{(\phi)}) = \sum_{i=1}^I \sum_{m=1}^{M_i} \rho_i^m(\mathbf{v}^{(\phi)}), \quad (3)$$

where  $\rho_i^m(\mathbf{v}^{(\phi)}) = (y_i^m - x_i^m)^2$ . In this case,  $x_i^m = \omega_i^m$  and  $y_i^m$  are the elements of input matrix  $\mathbf{X}_{csp}^1$  and output matrix  $\mathbf{Y}_{csp}^L$ , respectively. Then, we can compute the global gradient of using EDL at  $\phi$  by

$$\nabla \mathbf{v}^{(\phi)} = \frac{\partial \rho(\mathbf{v}^{(\phi)})}{\partial \mathbf{v}^{(\phi)}}. \quad (4)$$

After  $\nabla \mathbf{v}^{(\phi)}$  is obtained, the CSP updates the global model  $\mathbf{v}^{(\phi)}$  to minimize the prediction error, i.e.,  $\min_{\mathbf{v}} \rho(\mathbf{v})$ , using adaptive learning rate optimizer *Adam* [16] which produces fast convergence and significant robustness to the model. Let  $\eta_\phi$  and  $\delta_\phi$  denote the exponential moving average of the  $\nabla \mathbf{v}^{(\phi)}$  and the squared  $\nabla \mathbf{v}^{(\phi)}$  to infer the variance at  $\phi$ , respectively. Then, the update rules of  $\eta_{\phi+1}$  and  $\delta_{\phi+1}$  can be described as follows:

$$\begin{aligned} \eta_{\phi+1} &= \gamma_\eta^\phi \eta_\phi + (1 - \gamma_\eta^\phi) \nabla \mathbf{v}^{(\phi)}, \\ \delta_{\phi+1} &= \gamma_\delta^\phi \delta_\phi + (1 - \gamma_\delta^\phi) (\nabla \mathbf{v}^{(\phi)})^2, \end{aligned} \quad (5)$$

where  $\gamma_\eta^\phi$  and  $\gamma_\delta^\phi \in [0, 1]$  indicate the exponential decay steps of  $\eta_\phi$  and  $\delta_\phi$  at  $\phi$ , respectively. Furthermore, we consider the learning step  $\lambda$  to determine how fast we need to update the global model. Specifically, the update rule for  $\lambda$  can be expressed by

$$\lambda_{\phi+1} = \lambda \frac{\sqrt{1 - \gamma_\delta^{\phi+1}}}{1 - \gamma_\eta^{\phi+1}}. \quad (6)$$

Then, the global model  $\mathbf{v}_k^{(\phi+1)}$  to learn  $\mathbf{X}_{csp}$  for the next epoch time  $\phi + 1$  is updated as follows:

$$\mathbf{v}^{(\phi+1)} = \mathbf{v}^{(\phi)} - \lambda_{\phi+1} \frac{\eta_{\phi+1}}{\sqrt{\delta_{\phi+1} + \varepsilon}}, \quad (7)$$

where  $\varepsilon$  represents a constant to avoid zero division when the  $\sqrt{\delta_{\phi+1}}$  is almost zero. The learning process repeats and then terminates when the prediction error converges, or a certain number of epoch time  $T$  is reached. In this case, the final global model  $\mathbf{v}^*$  in the CSP is obtained to predict  $\hat{\mathbf{Y}}_{csp}$  of training dataset  $\mathbf{X}_{csp}$  and new dataset  $\hat{\mathbf{X}}_{csp}$  using Eq. (1). The algorithm for energy demand prediction using EDL is summarized in Algorithm 1. The processes between Lines 4 and 11 are implemented in the CSP.

### B. Federated Energy Demand Learning

Since the CSP needs to collect data from CSs, this centralized learning may lead to the communication overhead and data privacy concerns. To deal with these issues, we develop a framework using federated energy demand learning (FEDL) method. Particularly, the CSP only requires to collect the trained models, i.e., gradient information, from the set of CSs, and then updates the global model efficiently before sending back to the CSs [10]. After that, the CSs can use this information to learn by themselves using deep learning method. Given that  $\mathcal{J} = \{1, \dots, j, \dots, J\}$  as the set of CSs

### Algorithm 1 EDL-Based Prediction Algorithm

- 1: Initialize  $a_{csp}$  and  $\mathbf{v}^{(\phi)}$  when  $\phi = 0$
- 2: Generate  $\mathbf{X}_i, \forall i \in \mathcal{I}$
- 3: Send  $\mathbf{X}_i, \forall i \in \mathcal{I}$  to the CSP
- 4: Set  $\mathbf{X}_{csp}^1 \leftarrow \sum_{i \in \mathcal{I}} \mathbf{X}_i$  with  $e_i^m, \forall m \in \mathcal{M}_i, \forall i \in \mathcal{I}$
- 5: **while**  $\phi \leq T$  **and**  $\rho(\mathbf{v}^{(\phi)})$  do not converge **do**
- 6:   Learn  $\mathbf{X}_{csp}^1$  to obtain  $\mathbf{Y}_{csp}^L$  at layer- $L$  using  $\mathbf{v}^{(\phi)}$
- 7:   Calculate  $\rho(\mathbf{v}^{(\phi)})$  and  $\nabla \mathbf{v}^{(\phi)}$
- 8:    $\phi \leftarrow \phi + 1$
- 9:   Update  $\mathbf{v}^{(\phi)}$
- 10: **end while**
- 11: Predict  $\hat{\mathbf{Y}}_{csp}$  using  $\hat{\mathbf{X}}_{csp}$  and  $\mathbf{v}^*$  in the CSP for the next-period energy requests

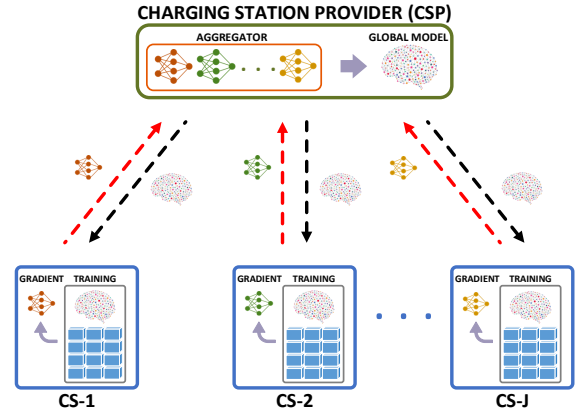


Fig. 2: Federated energy demand learning architecture.

which acts as workers to implement the EDL algorithms using their  $\mathbf{X}_j$  locally as illustrated in Fig. 2. Thus,  $\mathbf{X}_j$  is expressed by

$$\mathbf{X}_j = \sum_{i=1}^I \beta_j^i \mathbf{X}_i, \quad (8)$$

where  $\beta_j^i$  is a unique binary variable with  $\beta_j^i = 1$  indicating that the CS- $j$  has information from CS- $i$ , and  $\beta_j^i = 0$  otherwise.

For DNN, the output matrix of layer- $l$  at CS- $j$  is generated by

$$\mathbf{Y}_j^l = a_j(\mathbf{G}_l \mathbf{X}_j^l + \mathbf{h}_l), \quad (9)$$

where  $\mathbf{X}_j^l$  is the input matrix of layer  $l$  at CS- $j$  (with  $\mathbf{X}_j^1 = \mathbf{X}_j$  and  $a_j$  is the *tanh* activation function at CS- $j$ ). Additionally, we use dropout layer to eliminate the input  $\mathbf{X}_j^{l,drop}$  by a fraction rate  $f$ . At the output layer, we can obtain output matrix  $\mathbf{Y}_j^L$  and find the prediction error for each  $\phi$  at CS- $j$  as follows:

$$\rho_j(\mathbf{v}^{(\phi)}) = \sum_{i=1}^I \sum_{m=1}^{M_i} \beta_j^i \rho_i^m(\mathbf{v}^{(\phi)}). \quad (10)$$

Next, we can compute the local gradient at CS- $j$  by

$$\nabla \mathbf{v}_j^{(\phi)} = \frac{\partial \rho_j(\mathbf{v}^{(\phi)})}{\partial \mathbf{v}^{(\phi)}}. \quad (11)$$

Upon completing  $\nabla \mathbf{v}_j^{(\phi)}$  for every  $\phi$ , each CS- $j$  sends the

local gradient to the CSP for global gradient aggregation as expressed by

$$\nabla \mathbf{v}^{(\phi)} = \frac{1}{J} \sum_{j=1}^J \nabla \mathbf{v}_j^{(\phi)}. \quad (12)$$

In particular, the CSP acts as a model server to accumulate the gradients, and then updates the global model  $\mathbf{v}^{(\phi)}$  before sending back to the CS- $j$ ,  $\forall j \in \mathcal{J}$  (as illustrated in Fig. 2). This allows all CS- $j$ ,  $\forall j \in \mathcal{J}$  to collaborate by sharing local model information to each other to further improve the prediction accuracy through the CSP. To ensure that the gradient staleness is 0, the gradient aggregation is enabled immediately after  $J$  local gradients are received by the CSP synchronously. In this way, the gradient staleness occurs when the local gradients are computed using an obsolete global model.

To minimize the prediction error, i.e.,  $\min_{\mathbf{v}} \rho_j(\mathbf{v})$ , we also apply the Adam optimizer and update the global model  $\mathbf{v}^{(\phi+1)}$  as expressed in Eqs. (5)-(7). This  $\mathbf{v}^{(\phi+1)}$  is then pushed back to the CS- $j$ ,  $\forall j \in \mathcal{J}$  for the next local learning process. The aforementioned process continues until the prediction error converges or a certain number of epoch time  $T$  is reached. Then, we can predict  $\hat{\mathbf{Y}}_j, \forall j \in \mathcal{J}$  of training dataset  $\mathbf{X}_j, \forall j \in \mathcal{J}$  and new dataset  $\hat{\mathbf{X}}_j, \forall j \in \mathcal{J}$  at CS- $j$ ,  $\forall j \in \mathcal{J}$  using  $\mathbf{v}^*$  through Eq. (9). The algorithm for energy demand prediction using FEDL is shown in Algorithm 2 in which the processes between Lines 9 and 11 is executed in the CSP.

---

**Algorithm 2** FEDL-Based Prediction Algorithm

---

- 1: Initialize  $a_j$  and  $\mathbf{v}^{(\phi)}$  when  $\phi = 0$
  - 2: Generate  $\mathbf{X}_j, \forall j \in \mathcal{J}$
  - 3: **while**  $\phi \leq T$  **and**  $\rho_j(\mathbf{v}^{(\phi)}), \forall j \in \mathcal{J}$  do not converge **do**
  - 4:   **for**  $\forall j \in \mathcal{J}$  **do**
  - 5:     Learn  $\mathbf{X}_j^1$  to obtain  $\mathbf{Y}_j^L$  at layer- $L$  using  $\mathbf{v}^{(\phi)}$
  - 6:     Calculate  $\rho_j(\mathbf{v}^{(\phi)})$  and  $\nabla \mathbf{v}_j^{(\phi)}$
  - 7:     Send  $\nabla \mathbf{v}_j^{(\phi)}$  to the CSP for global model update
  - 8:   **end for**
  - 9:   Compute current  $\mathbf{v}^{(\phi)}$
  - 10:    $\phi \leftarrow \phi + 1$
  - 11:   Update and send  $\mathbf{v}^{(\phi)}$  back to  $J$  CSs
  - 12: **end while**
  - 13: **for**  $\forall j \in \mathcal{J}$  **do**
  - 14:   Predict  $\hat{\mathbf{Y}}_j$  using  $\hat{\mathbf{X}}_j$  and  $\mathbf{v}^*$  for the next-period energy requests
  - 15: **end for**
- 

### C. Clustering-Based Energy Demand Learning

Learning the dataset without considering the useful feature classification as the aforementioned approaches may produce biased energy demand prediction, especially when we combine imbalanced features and known labels in one dataset. Then, to obtain better prediction accuracy, we can group CSs into  $K$  clusters before the learning process is performed. In this case, the clustering decision is determined by using location information of the CSs, i.e., the latitude and longitude. We

utilize the constrained K-means algorithm [17] to perform the clustering process. In particular, we modify the constrained K-means algorithm to generate balance number of the CSs in each cluster through using minimum and maximum cluster size constraints. This consideration is to give a fairness of the learning process for the CSs in each cluster based on their deployment locations. Given  $\mathcal{K} = \{1, \dots, k, \dots, K\}$  as the set of clusters and dataset  $\mathbf{N}$  containing CS IDs and their locations  $n_i, \forall i \in \mathcal{I}$ , we aim to determine centroids (i.e., cluster centers)  $C_k, \forall k \in \mathcal{K}$  such that the accumulation of squared distance between each  $n_i$  and its closest centroid  $C_k$  is minimized as optimization problem described below.

$$\min_{\{\tau, \mathbf{C}\}} \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}} \tau_i^k (n_i - C_k)^2, \quad (13)$$

$$\text{s.t. } \theta_k^{low} \leq \sum_{i \in \mathcal{I}} \tau_i^k \leq \theta_k^{high}, \forall k \in \mathcal{K}, \quad (14)$$

$$\sum_{k \in \mathcal{K}} \tau_i^k = 1, \forall i \in \mathcal{I}, \quad (15)$$

$$\tau_i^k \in \{0, 1\}, \forall i \in \mathcal{I}, \forall k \in \mathcal{K}, \quad (16)$$

$$\theta_{low}^k \geq 0, \theta_{high}^k \geq 0, \forall k \in \mathcal{K}, \quad (17)$$

where  $\tau_i^k$  is a binary variable with  $\tau_i^k = 1$  representing that the location of CS- $i$  is the nearest to the centroid  $C_k$  and  $\tau_i^k = 0$  otherwise. The constraints (14) guarantee that the CSs' locations in each cluster- $k$  are within predefined minimum  $\theta_{low}^k$  and maximum  $\theta_{high}^k$  thresholds. Additionally, the constraints (15) indicate that a location of CS- $i$  is clustered to the closest centroid only. To reach the optimal solution, we need to update the centroid  $C_k^{(t)}$  at each iteration  $t$ . Specifically, in the cluster- $k$ , we have

$$C_k^{(t+1)} = \frac{\sum_{i \in \mathcal{I}} \tau_i^{k,(t)} n_i}{\sum_{i \in \mathcal{I}} \tau_i^{k,(t)}}, \quad (18)$$

if  $\theta_{low}^k \leq \sum_{i \in \mathcal{I}} \tau_i^k \leq \theta_{high}^k$  and

$$C_k^{(t+1)} = C_k^{(t)}, \quad (19)$$

otherwise. The process terminates when  $C_k^{(t+1)} = C_k^{(t)}, \forall k \in \mathcal{K}$ , and thus we can obtain the optimal set  $\mathcal{I}_k$  in the cluster- $k$ . To this end, we can perform the centralized EDL or FEDL method to predict the energy demand for the CSs in each cluster independently. The algorithm for the clustering-based EDL using customized constrained K-means method is shown in Algorithm 3.

## IV. PERFORMANCE EVALUATION

### A. Dataset Pre-Processing and Evaluation Method

To evaluate the performance of the proposed learning methods, we use the real data obtained from charging stations in Dundee city, the United Kingdom between 2017 and 2018 [13]. In particular, the dataset has 65,601 transactions which include CS ID from 58 CSs, transaction ID for each CS, EV charging date, EV charging time, and consumed

---

**Algorithm 3** Clustering-Based EDL with Customized Constrained K-Means Algorithm
 

---

- 1: Generate  $\mathcal{N}$  containing  $n_i, \forall i \in \mathcal{I}$
  - 2: Set  $K$  and initialize random  $C_k, \forall k \in \mathcal{K}$  when  $t = 0$
  - 3: **while**  $C_k^{(t+1)} \neq C_k^{(t)}, \forall k \in \mathcal{K}$  **do**
  - 4:   Solve optimization problem in Eq. (13) with the constraints in Eqs. (14)-(17)
  - 5:   Update  $t + 1$
  - 6:   **for**  $\forall k \in \mathcal{K}$  **do**
  - 7:     **if**  $\theta_{low}^k \leq \sum_{i \in \mathcal{I}} \tau_i^k \leq \theta_{high}^k$  **then**
  - 8:       
$$C_k^{(t+1)} = \frac{\sum_{i \in \mathcal{I}} \tau_i^{k,(t)} n_i}{\sum_{i \in \mathcal{I}} \tau_i^{k,(t)}}$$
  - 9:     **else**
  - 10:        $C_k^{(t+1)} = C_k^{(t)}$
  - 11:     **end if**
  - 12:   **end for**
  - 13: **end while**
  - 14: Generate the optimal set  $\mathcal{I}_k$  in the cluster- $k, \forall k \in \mathcal{K}$
  - 15: Perform EDL or FEDL method using Algorithm 1 or 2, respectively, in the cluster- $k, \forall k \in \mathcal{K}$
- 

energy (in kWh) for each transaction. We use the first four information as the learning features, and the consumed energy as the learning label. Then, we classify CS ID, charging date, and charging time as categorical features. Specifically, we convert the charging date and charging time information into 7-day (i.e., 1, 2, ..., 7) and 24-hour (i.e., 0, 1, ..., 23) categories, respectively. In addition, each CS has the latitude and longitude information which will be used for clustering.

Then, we adopt the RMSE to show the prediction accuracy, i.e., prediction error, because we deal with the prediction of energy demand which is categorized as a regression prediction model, i.e., when the mapping function yields the continuous prediction outputs. Given  $S$  transactions, the RMSE can be computed as follows:

$$\text{RMSE} = \sqrt{\frac{1}{S} \sum_{s=1}^S (\omega_s - \hat{\omega}_s)^2}, \quad (20)$$

where  $\omega_s$  and  $\hat{\omega}_s$  are the actual and predicted energy demand for transaction  $s$ .

### B. Experimental Setup

We evaluate the performance using *TensorFlow CPU* in UTS shared cluster with configuration Intel Xeon E5-2687W v2 3.4GHz 8 cores 32GB RAM. We compare our proposed methods with some other conventional machine learning methods including decision tree (DT), random forest (RF), support vector regressor (SVR), k-neighbors regressor (KNN), stochastic gradient descent regressor (SGDR), and multi-layer perceptron regressor (MLPR) [18]. We split the dataset into 80%, 70%, 60%, as well as 50% training dataset, and the rest of the portions for testing dataset. From the training dataset, we divide the number of transactions by  $J$  training subsets, when FEDL is implemented. Each CS- $j$  runs the testing dataset for

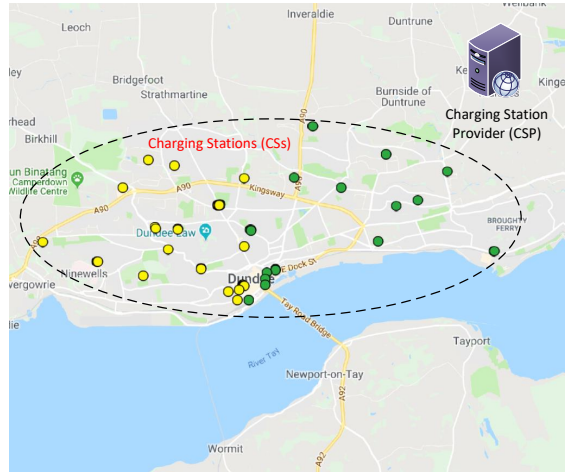


Fig. 3: Charging station clustering.

the upcoming energy demand prediction. For clustering, we define  $K = 2$ , and thus we divide 58 CSs into 2 clusters as shown in Fig. 3. For DNN, we use two hidden layers with 64 neurons per layer and one dropout layer with a fraction rate 0.15. We also apply the adaptive learning rate Adam optimizer with initial step size 0.01 and *tanh* function as the activation function.

### C. Simulation Results

Table I shows the comparisons between baseline and proposed machine learning methods for various training set ratios. We first evaluate the prediction accuracy, i.e., RMSE, of the testing set when 80% training set ratio is used. In particular, the RMSE obtained by the centralized EDL with clustering and the FEDL with clustering are 24.28% and 24.63% lower than those of the baseline methods, respectively. The reason is that the clustering approach can improve the prediction accuracy by grouping similar useful features and/or labels together in the same cluster [12]. As such, by clustering the CSs based on their locations, we can reduce the cost of biased prediction results over the whole dataset, which leads to the lower prediction error. For the FEDL and the centralized EDL without clustering, the RMSE performance is also always better than those of the baseline learning methods up to 23.51%. Moreover, the gap between the FEDL as well as the centralized EDL without clustering and the proposed learning methods with clustering is only within 2%. This is because the FEDL and the centralized EDL can deeply learn the useful features from the dataset. Specifically, the use of hyperparameter settings in the DNN, e.g., the number of hidden layers and neurons, the regularization methods, the activation functions, and the mini-batch size, contributes to the prediction accuracy improvement. Additionally, for the FEDL, it can learn the subset of the whole dataset independently at different workers, and achieve the average prediction with less variance and lower error in respect of the number of workers.

In contrast to the proposed learning methods, the baseline learning methods cannot learn the useful features and their correlations deeply. The reason is that they do not explore



TABLE I: Testing RMSE of energy demand prediction for various learning methods and training set ratios

Learning approach	Training set ratio			
	80%	70%	60%	50%
KNR	7.18	7.71	7.57	7.67
MLPR	6.57	6.62	6.90	6.53
SGDR	6.55	6.57	6.54	6.54
DT	6.47	6.49	6.47	6.47
SVR	6.46	6.50	6.50	6.53
RF	6.35	6.66	6.88	6.80
<b>EDL</b>	5.86	5.86	5.86	5.87
<b>FEDL</b>	5.81	5.82	5.81	5.84
<b>EDL + Clustering</b>	<b>5.77</b>	<b>5.79</b>	5.85	5.87
<b>FEDL + Clustering</b>	<b>5.76</b>	<b>5.78</b>	<b>5.78</b>	<b>5.83</b>

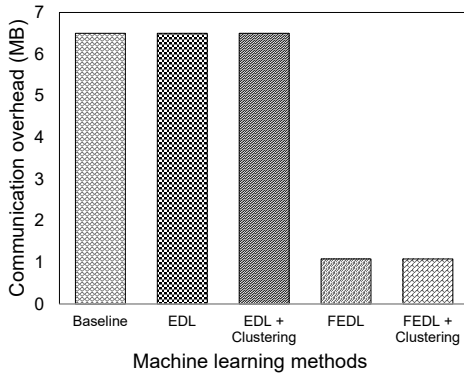


Fig. 4: Communication overhead between baseline and proposed machine learning methods.

nonlinear transformations of the complex hidden features and multiple levels of processing layers as provided by the EDL and FEDL methods. For other training set ratios (i.e., 0.7%, 0.6%, and 0.5%), we observe that the proposed learning methods still outperform all the baseline learning methods. In particular, the centralized EDL with clustering and the FEDL with clustering still have the lowest RMSE for training set ratio 0.7%. Furthermore, the FEDL with clustering has the lowest RMSE for the rest of training set ratios. This interesting trend can provide useful information for the CSP to choose the suitable proposed learning methods in terms of stability, robustness, and flexibility.

In Fig. 4, we show that the use of FEDL method can significantly reduce the communication overhead up to 83.4% compared with the centralized methods including all baseline methods and the centralized EDL. This is because the CSP only requires to collect the trained models from the workers without sending any real dataset. This advantage aligns with the reduction of privacy issue for participating EVs and CSs.

## V. CONCLUSION

In this paper, we have proposed the novel machine learning-based approaches leveraging deep learning techniques to improve the energy demand prediction accuracy and reduce the communication overhead in the EV network. In the first

approach, the CSP gathers information from all CS and then predict energy demand for the whole network using the deep learning technique. After that, we have introduced the federated energy demand learning-based method in which the energy demand learning can be performed at the CSs without disclosing the privacy of EVs and CSs. Furthermore, we have applied the clustering-based energy demand learning method for the CSs to further improve the energy demand prediction accuracy. Through the simulation results, we have demonstrated that our proposed approaches outperform other machine learning algorithms in terms of the prediction accuracy and communication overhead.

## REFERENCES

- [1] The Growth of Electric Vehicles. Available Online: <https://www.cnbc.com/2018/05/30/electric-vehicles-will-grow-from-3-million-to-125-million-by-2030-iaa.html>. Last Accessed on April 2019.
- [2] W. Tang, S. Bi, and Y. J. A. Zhang, "Online coordinated charging decision algorithm for electric vehicles without future information," *IEEE Transactions on Smart Grid*, vol. 5, no. 6, pp. 2810-2824, Nov. 2014.
- [3] J. A. P. Lopes, F. J. Soares, and P. M. R. Almeida, "Integration of electric vehicles in the electric power system," *Proc. IEEE*, vol. 99, no. 1, pp. 168183, Jan. 2011.
- [4] P. You and Z. Yang, "Efficient optimal scheduling of charging station with multiple electric vehicles via V2V," in *IEEE SmartGridComm*, Nov. 2014, pp. 716721.
- [5] M. Majidpour, *et al.*, "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, Feb. 2015.
- [6] A. Chis, J. Lunden, and V. Koivunen, "Reinforcement learning-based plug-in electric vehicle charging with forecasted price," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 5, pp. 3674-3684, May 2017.
- [7] A. Fukushima, *et al.*, "Prediction of energy consumption for new electric vehicle models by machine learning," *IET Intelligent Transport Systems*, vol. 12, no. 9, pp. 1174-1180, Oct. 2018.
- [8] W. Ma, V. Gupta, and U. Topcu, "Distributed charging control of electric vehicles using online learning," *IEEE Transactions on Automatic Control*, vol. 62, no. 10, pp. 5289-5295, Oct. 2017.
- [9] K. L. Lopez, C. Gagne, and M. Gardner, "Demand-side management using deep learning for smart charging of electric vehicles," *IEEE Transactions on Smart Grid*, 2019.
- [10] J. Dean *et al.*, "Large scale distributed deep networks," in *ACM NIPS 2012*, Dec. 2012, pp. 1223-1231.
- [11] Y. He, K. M. Kockelman, and K. A. Perrine, "Optimal locations of US fast charging stations for long-distance trip completion by battery electric vehicles," *Journal of Cleaner Production*, vol. 214, pp.452-461, Mar. 2019.
- [12] W. Li, *et al.*, "Implemented IoT-based self-learning home management system (SHMS) for Singapore," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2212-2219, Jun. 2018.
- [13] Electric Vehicle Charging Sessions Dundee. Available Online: <https://data.dundee.gov.uk/dataset/ev-charging-data>. Last Accessed on April 2019.
- [14] S. Romn, *et al.*, "Regulatory framework and business models for charging plug-in electric vehicles: infrastructure, agents, and commercial relationships," *Energy Policy*, vol. 39, no. 10, pp. 6360-6375, Oct. 2011.
- [15] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: a survey," arXiv:1803.04311 [cs.NI], Sept. 2018.
- [16] D. Kingma and J. Ba, "Adam: a method for stochastic optimization," in *ICLR 2015*, May 2015, pp. 1-15.
- [17] P. S. Bradley, K. P. Bennett, and A. Demiriz, "Constrained K-means clustering," *Microsoft Research MSR-TR-2000-65*, May 2000.
- [18] R. Boutaba, *et al.*, "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities," *Journal of Internet Services and Applications*, vol. 9, no. 1, pp. 1-99, Jun. 2018.