Federated Learning for Electric Vehicle Energy Demand Prediction Across Distributed Users

Vasilis Perifanis[1⋆][0000−0003−3915−9628], Foteini Nikolaidou[1][0009−0002−5691−9510], Panagiotis Thomakos[1][0009−0009−0296−8879], Nikolaos Pavlidis[1][0000−0001−9370−5023], Andreas Sendros[1][0000−0002−6337−4320], Ignacio Lacalle Úbeda[2][0000−0002−6002−4050], Felix Mondl[3], and Michael Glensvig[4]

[1] Indigma Innovations, Greece
{vasilis,f.nikolaidou,p.thomakos,nikos,andreas}@indigma.eu
[2] Universitat Politècnica de València, Spain
iglaub@upv.es
[3] Österreichische Post AG, Austria
felix.mondl@post.at
[4] AVL List GmbH, Austria
Michael.Glensvig@avl.com

Abstract. Accurate prediction of daily electric vehicle (EV) energy demand supports charging planning across home users, commercial fleets, depots, and charging service providers. We study this problem as a supervised regression task under centralized and Federated Learning (FL) settings, using heterogeneous EV data sources that include two public datasets and a private commercial dataset from a postal service. Raw EV records are transformed into vehicle-day samples through a unified preprocessing pipeline based on a common schema of vehicle identifier, date, time, state of charge (SOC), and odometer readings. Daily travelled distance and energy demand are derived from these records, and the prediction target is estimated from discharge segments between charging events. The input features also combine calendar, historical demand, and weather descriptors. We evaluate local, centralized, cross-device, and cross-silo FL across three deep learning architectures, and implement a buffered asynchronous FL orchestration pipeline on a Raspberry Pi testbed to assess practical deployment. The results show that distributed EV energy demand models can approach centralized performance while preserving data locality, and that buffered asynchronous FL is practical for heterogeneous EV deployments.

Keywords: Electric Vehicles · Energy Demand Prediction · Federated Learning · Buffered Asynchronous FL · Privacy-Preserving AI.

1 Introduction

The increasing adoption of electric vehicles (EVs) is reshaping energy demand planning at the user, fleet, and infrastructure levels [25]. In all these settings,

⋆ Corresponding author.

knowing how much energy an EV is expected to require over a future day supports better charging decisions, reduces operational uncertainty, and enables more informed use of available charging capacity [10].

A practical forecasting task in this context is the prediction of EV energy demand from expected daily mobility. In this setting, the goal is to estimate the total energy an EV will require over a given day from planning-level information, such as the anticipated distance to be travelled. This formulation is directly aligned with real-world planning: a home user may know the approximate distance to be travelled the next day, or a fleet or depot operator may have access to route plans or vehicle schedules. If the corresponding energy requirement can be estimated in advance, charging can be planned earlier and more effectively.

This prediction supports multiple use cases. For home users, daily energy demand estimates help determine whether overnight charging is sufficient, whether charging should be delayed to lower-cost periods, or whether locally generated energy can cover part of the expected need. For fleet operators, the same prediction supports charger allocation, depot readiness, and operational continuity. For charging service providers and aggregators, daily demand estimates contribute to short-term load planning. In the longer term, such predictions may be combined with photovoltaic generation forecasts, local storage, electricity prices, and charger constraints to enable energy-aware charging schedules [1].

Learning reliable EV energy demand models is challenging because EV data are naturally distributed. Different EVs, users, and fleets exhibit different usage patterns, sampling characteristics, and operating conditions. Public EV datasets provide reproducible benchmarks but typically cover a limited number of vehicles or operating scenarios [23,16]. Private EV datasets offer valuable real-world coverage but are often restricted by privacy, governance, or commercial confidentiality constraints. These limitations make centralized model training difficult in settings where collaboration across users or organizations would be beneficial.

Several challenges arise in this setting. EV data are non-IID and often imbalanced across vehicles or organizations, telemetry may include missing values, SOC noise, odometer resets, and irregular sampling, and deployment must account for privacy, governance, intermittent availability, and edge-resource constraints [22]. These challenges motivate a Federated Learning (FL) formulation for heterogeneous data ownership and uneven client data sizes.

FL offers a natural solution to this problem [11]. In FL, participating clients train models locally and share only model updates with a coordinator server, without transmitting raw data. This allows a shared model to be learned across distributed sources while preserving data locality. For EV energy demand prediction, each client may correspond to a vehicle, user, household, depot, or fleet, making FL well suited to the broader EV charging scenarios where data ownership and privacy are important [14].

In this work, we study EV energy demand prediction as a supervised task under centralized and FL settings. Each sample corresponds to one vehicle-day. Raw EV records are transformed into daily driving events using a common schema, from which daily travelled distance and energy demand are derived. The learning objective is to estimate the required daily energy from the expected distance, calendar features, weather descriptors, and historical demand summaries.

We evaluate the proposed formulation using two public EV datasets [23,16] and a private commercial dataset from a postal service. We compare local, centralized, cross-device, and cross-silo FL across three deep learning (DL) architectures. Beyond predictive accuracy, we also consider the practical deployment of FL, since real systems must address communication cost, latency, client availability, and update synchronization [22]. To this end, we implement a buffered asynchronous FL orchestration pipeline on a Raspberry Pi testbed, where the server aggregates received updates once a predefined buffer condition is satisfied [13]. This design is well suited to EV settings, where vehicles or fleets may differ in local data size, computational capacity, and communication delay.

The main contributions of this work are summarized as follows:

1. We formulate daily EV energy demand prediction as a planning-level regression task over vehicle-day samples using expected distance, calendar, weather, and historical demand features.
2. We build a unified preprocessing pipeline that maps heterogeneous public and private EV records to a common schema and derives daily energy-demand targets.
3. We compare centralized, cross-device, and cross-silo FL across three model architectures, showing that FL can approach centralized performance while preserving data locality.
4. We deploy buffered asynchronous FL on a Raspberry Pi testbed and report accuracy together with communication, latency, update-delay, and convergence metrics.

This work is developed in the context of the O-CEI project[5], where FL is considered as an enabling mechanism for cross-domain collaboration without centralizing raw data. The O-CEI marketplace[6] provides a potential deployment channel for sharing trained models, generated datasets, or updated model weights; however, the scientific focus of this paper is the EV energy-demand formulation, preprocessing pipeline, FL evaluation, and edge deployment.

The remainder of this paper is organized as follows. Section 2 reviews related work on EV energy demand prediction and FL Section 3 describes the dataset construction, feature engineering, and learning formulation. Section 4 presents the experimental settings and results. Section 5 concludes the paper.

2 Related Work

EV energy prediction has been studied from several complementary perspectives, including vehicle energy consumption modelling, remaining range estimation, charging load forecasting, and privacy-preserving distributed learning. Recent

[5] `https://o-cei.eu/`
[6] `https://marketplace.pre.o-cei.eu/`

reviews the growing role of machine learning (ML) and DL methods, but also show that most formulations focus on trip-level consumption, range prediction, SOC management, or charging infrastructure integration [25,2,18].

Recent studies have advanced EV energy consumption prediction using real-world operating, temporal, environmental, and driver-specific factors. Hussain et al. [7] study real-world EV energy consumption prediction, Wang et al. [21] propose a transformer-based model for trip-level battery EV consumption under inconsistent feature spaces, and Rachavelpula and Cha [17] combine map-derived context, driver-specific velocity prediction, and physics-based modelling for personalized energy estimation. Public datasets such as Dynamic BEV [23] and ESOGU-ML5EV [16] support reproducible benchmarking, but remain limited in vehicle coverage or operating contexts, motivating approaches that can combine public and private data without centralizing raw records.

A second line of work focuses on charging demand and station-level load forecasting, where the target is commonly aggregate station load, regional demand, or charging behaviour. Van Etten et al. [5] study large-scale charging demand forecasting using time-series models and evaluate generalization to unseen stations. Gowda et al. [12] apply transfer learning and model-agnostic meta-learning for charging load forecasting. Tian et al. [19] combine temporal convolutional networks and LSTM for short-term charging load prediction, while Chen et al. [4] examine charging station demand under climatic influence. Cavus et al. [3] forecast EV charging demand in smart city settings using behavioural, spatial, and demographic data. These studies address grid operation and infrastructure planning, but their targets are aggregated charging load or station-level usage rather than the daily energy required by a specific EV given expected mobility.

FL has recently gained visibility in EV charging and energy analytics, as EV and charging records are naturally distributed across users, vehicles, stations, fleets, and organizations. Yin and Ji [24] propose a horizontal FL framework for EV charging load prediction using VMD-LSTM models. Han and Li [6] study vertical FL for charging station load prediction in coupled transportation and power distribution systems. Jin and Liu [8] propose a blockchain-based federated transfer learning framework for privacy-aware probabilistic EV load forecasting. Marlin et al. [9] introduce a hierarchical FL framework for spatio-temporal EV charge prediction with secure aggregation. Perifanis et al. [15] study FL for early prediction of EV charging demand, where the total session energy is estimated using information available at plug-in time and during the first minutes of charging. Tritsarolis et al. [20] examine the effect of FL on EV energy demand forecasting across distributed users. While these works demonstrate the applicability of FL to EV and charging analytics, most existing formulations remain station-oriented, grid-facing, session-level, or next-charge oriented. In contrast, the present work studies planning-level daily EV energy demand prediction from vehicle-day samples derived from telemetry and SOC trajectories, and evaluates buffered asynchronous FL on a real edge hardware testbed.

Table 1. Dataset-specific battery capacities and weather sources.

Dataset group	Vehicles	Battery capacity (kWh)	Weather source
Dynamic BEV 1 [23]	1	60.5	Genk, Belgium
Dynamic BEV 2 [23]	1	78.8	Genk, Belgium
ESOGU [16]	1	15.5	Eskişehir, Turkey
Private postal operations	10	50.0	Local weather data

3 Dataset Construction and Learning Formulation

3.1 Problem Definition

We formulate daily EV energy demand prediction as a supervised regression task. Let v denote an EV and d a calendar day. Each vehicle-day pair defines one sample with input vector $\mathbf{x}_{v,d}$ and target $y_{v,d}$, corresponding to the energy required by vehicle v at day d, expressed in kWh. The learning objective is to estimate $f : \mathbf{x}_{v,d} \mapsto y_{v,d}$. The input vector combines planning, temporal, historical, and weather features, with EV embeddings included as a learned branch. The main planning variable is the expected daily distance, which in offline experiments is approximated from the realized distance through binning, and in deployment can be replaced by a user-provided estimate, or a schedule.

We evaluate four learning settings. In the *local* setting, one model is trained independently per dataset group. In the *centralized* setting, all training samples are pooled. In the *cross-device* FL setting, each EV acts as one client. In the *cross-silo* FL setting, each dataset group acts as one client, representing collaboration across data owners with heterogeneous operational contexts.

3.2 Preprocessing pipeline

The original datasets differ in structure and naming conventions. All sources are first mapped to a common minimal schema $\{\texttt{EV id}, \texttt{date}, \texttt{time}, \texttt{SOC}, \texttt{odometer}\}$. For energy estimation, we use dataset-specific nominal battery capacities. The two Dynamic BEV vehicles [23] use 60.5 kWh and 78.8 kWh, respectively, the ESOGU [16] EV uses 15.5 kWh, and the private postal-operation EVs use 50.0 kWh. Weather information is associated with each dataset group according to its location. Table 1 summarizes the configuration.

After schema harmonization, records are ordered by EV identifier and timestamp. The odometer signal is repaired before distance computation, since real telemetry may include backward jumps or reset artifacts. To avoid negative travelled distances, the odometer is forced to be non-decreasing within each vehicle trajectory via a cumulative maximum. The SOC signal is cleaned by treating values outside $[0, 100]$ as invalid and smoothing isolated one-step bounces where the previous and next values are equal and the middle value differs only slightly. Interval-level differences are then computed between consecutive records. Let

$\Delta O_{v,t}$ denote the odometer difference and B_v the nominal battery capacity of vehicle v. The battery energy at timestamp t is approximated as:

$$E_{v,t} = \frac{\mathrm{SOC}_{v,t}}{100} B_v, \tag{1}$$

and positive discharge between consecutive records is defined as:

$$\Delta E^{\mathrm{dis}}_{v,t} = \max\left(E_{v,t-1} - E_{v,t}, 0\right). \tag{2}$$

When $\Delta O_{v,t} > 0$, an interval-level consumption estimate is computed as:

$$c_{v,t} = \frac{\Delta E^{\mathrm{dis}}_{v,t}}{\Delta O_{v,t}}. \tag{3}$$

Cleaned interval-level data are aggregated into a vehicle-day table. For each vehicle-day, we compute the total travelled distance, starting, minimum and ending SOC, and summary statistics of interval-level consumption.

The daily energy target is estimated using discharge segments between charging events. A charging event is detected as a contiguous run of positive SOC increments with limited odometer movement, retained when the SOC gain is at least 3 percentage points and the odometer gain is at most 1 km. This captures parked charging events and avoids treating driving-related SOC fluctuations as charging. Daily discharged energy is estimated by summing SOC drops between charging runs, with each segment's contribution computed from the difference between the segment peak SOC and the following valley SOC, multiplied by the nominal battery capacity. The resulting target is:

$$y_{v,d} = \{\texttt{day energy kwh}\}_{v,d}. \tag{4}$$

The expected daily distance is represented through a coarse planning proxy. The realized daily distance $D_{v,d}$ is rounded down to the nearest 10 km bin:

$$\{\mathrm{expectedkm}\}_{v,d} = 10 \left\lfloor \frac{\max(D_{v,d}, 0)}{10} \right\rfloor, \tag{5}$$

preserving the planning-level nature of the task while avoiding the use of exact realized distance as a direct predictor. Calendar variables (month, ISO week, weekday, and a weekend indicator) are extracted and encoded with sine and cosine transformations to capture periodic structure.

Weather records are aggregated at the daily level. Temperature features include daily mean, minimum, maximum, and standard deviation, together with morning and typical driving-hour summaries, as well as heating and cooling degree indicators relative to base temperatures of 18°C and 22°C. Precipitation features include total daily precipitation, maximum and mean hourly precipitation, precipitation during typical driving hours, total precipitation duration, and binary indicators for rainy, heavy-rain, morning-rain, and afternoon-rain days.

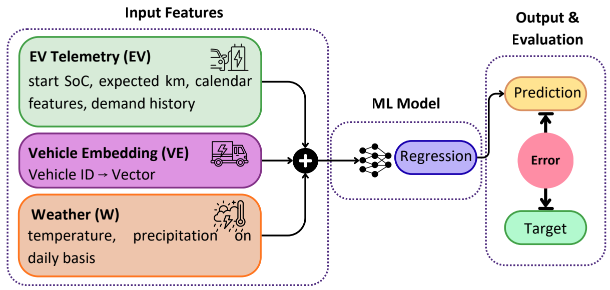


Fig. 1. Pipeline for daily EV energy demand prediction. EV telemetry provides the core daily input. Learned EV embeddings and weather descriptors are fused with the telemetry representation and processed by a deep tabular model.

Same-day weather variables are used as forecast inputs; in a production setting, these values would be obtained from day-ahead or intra-day weather forecasts, whereas in offline experiments historical observations are used as proxy forecasts. Access to weather information is assumed to be feasible for all clients, since EVs can obtain area-level weather data either through onboard sensing systems or via network connectivity when visiting a depot [7].

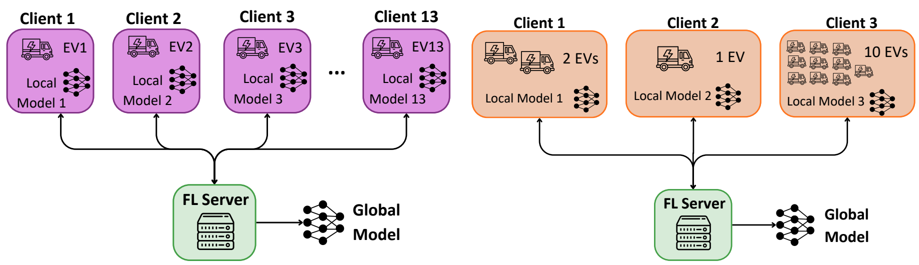
After daily aggregation and weather merging, each sample has the form $(\mathbf{x}_{v,d}, y_{v,d})$, where $y_{v,d}$ is the daily energy demand. The base feature set is:

$$\begin{aligned} \mathbf{x}_{v,d}^{\text{base}} = \{&\texttt{start SOC}, \texttt{expected km}, \texttt{month sin}, \texttt{month cos}, \\ &\texttt{week sin}, \texttt{week cos}, \texttt{weekday sin}, \texttt{weekday cos}, \texttt{is weekend}\}. \end{aligned} \tag{6}$$

To capture recent EV-specific usage patterns, we include historical energy demand features computed from past days of the same vehicle. The first is the previous day's energy demand, $y_{v,d-1}$, and the second is the average energy demand over the previous three available days, $\bar{y}_{v,d}^{(3)} = \frac{1}{m_{v,d}} \sum_{j=1}^{m_{v,d}} y_{v,d-j}$, where $m_{v,d} = \min(3, N_{v,d}^{\text{past}})$ and $N_{v,d}^{\text{past}}$ denotes how many past observations are available for vehicle v before day d. Weather features are appended to complete the input vector. Rows with missing predictors or targets are removed before training. The dataset is split into training, validation, and test subsets within each vehicle, with feature and target scaling estimated on the training split. Fig. 1 illustrates the pipeline. Vehicle telemetry provides the mandatory daily signal. Learned vehicle embeddings and weather descriptors are fused with the telemetry representation before the regression head predicts the daily energy target.

3.3 Federated Learning Settings

We evaluate FL under two partitioning schemes (Fig. 2). In the *cross-device* setting, each EV is one client, yielding 13 clients in total (two Dynamic BEV, one ESOGU EV, and 10 private EVs). In the *cross-silo* setting, each dataset group is one client, yielding three clients. The clients correspond to heterogeneous groups collected in different locations and operational contexts.



(a) Cross-device FL: one client per EV (13 clients total).

(b) Cross-silo FL: one client per dataset group (3 clients total).

Fig. 2. Federated partitioning schemes. In both cases, the FL server exchanges only model parameters with clients. Cross-device FL captures vehicle-level heterogeneity; cross-silo FL models collaboration across data silos.

Algorithm 1 Buffered Asynchronous FedAvg

Require: Initial model w^0, clients $\mathcal{K} = \{1, \ldots, K\}$, maximum server updates T, local epochs E, learning rate η, buffer size B, staleness limit $\tau_{\max}$

Initialize server update index $t \leftarrow 0$ and buffer $\mathcal{B} \leftarrow \emptyset$
while $t < T$ **do**
  Client polling and local training
  for all available clients k that poll the server **do**
    Server returns the current task (w^t, t) to client k
    Client k sets $w_k \leftarrow w^t$ and $t_k \leftarrow t$
    for $e = 1, \ldots, E$ **do**
      Client k updates w_k on local data $\mathcal{D}_k$ using learning rate η
    end for
    Client k submits update (w_k, n_k, t_k) to the server
  end for
  Server-side buffering
  for all submitted updates (w_k, n_k, t_k) **do**
    if $t - t_k \leq \tau_{\max}$ **then**
      $\mathcal{B} \leftarrow \mathcal{B} \cup \{(w_k, n_k, t_k)\}$
    end if
  end for
  Buffered aggregation
  if $|\mathcal{B}| \geq B$ **then**
    Select a batch $\mathcal{U}^t \subseteq \mathcal{B}$ with $|\mathcal{U}^t| = B$
    $n_{\mathcal{U}} \leftarrow \sum_{(w_k, n_k, t_k) \in \mathcal{U}^t} n_k$
    $w^{t+1} \leftarrow \sum_{(w_k, n_k, t_k) \in \mathcal{U}^t} \frac{n_k}{n_{\mathcal{U}}} w_k$
    $\mathcal{B} \leftarrow \mathcal{B} \setminus \mathcal{U}^t$
    $t \leftarrow t + 1$
  end if
end while
return w^T

Buffered Asynchronous FedAvg The global FL objective is:

$$F(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w), \tag{7}$$

where $n = \sum_{k=1}^{K} n_k$ and the objective for client k with dataset $\mathcal{D}_k$ of size n_k is:

$$F_k(w) = \frac{1}{n_k} \sum_{(\mathbf{x}_i, y_i) \in \mathcal{D}_k} \ell(f_w(\mathbf{x}_i), y_i), \tag{8}$$

and $\ell(\cdot)$ is the regression loss.

To support heterogeneous and intermittently available clients, we adopt buffered asynchronous FedAvg [13]. Unlike synchronous FL, the server does not wait for all selected clients before aggregating. Instead, client updates are stored as they arrive and the global model is updated once the buffer reaches a predefined size. This setting reduces idle waiting time and is well suited to EV deployments where vehicles, users, or organizational silos may differ in data volume, connectivity, and computational capacity. The procedure is detailed in Algorithm 1.

4 Experiments

4.1 Dataset

The experiments use the final vehicle-day dataset produced by the preprocessing pipeline described in Section 3. The dataset contains 3247 samples, split randomly within each vehicle into 2271 training, 488 validation, and 488 test samples. Table 2 reports the resulting counts and shows the client definitions used in the centralized, cross-device, and cross-silo settings. The data are highly imbalanced across sources. The private postal operation group contributes 2864 samples (88.2% of the full dataset), Dynamic BEV contributes 379 samples, and ESOGU contributes 4 valid vehicle-day samples after preprocessing. At the vehicle level, the postal EVs contain between 247 and 312 samples each, while the public vehicles range from 4 samples to 256. The cross-silo setting therefore represents collaboration across data owners with substantially different data sizes.

Fig. 3 summarizes the distribution of daily consumption (kWh/km). The private EVs exhibit the widest spread and several high-consumption outliers, while Dynamic BEV and ESOGU show lower medians and narrower interquartile ranges. Despite overlapping consumption ranges, individual EVs display distinct density profiles, confirming the heterogeneous nature of the FL setting. Fig. 4 shows the temporal variation of daily consumption, with a seasonal increase toward winter months followed by a decrease toward spring, consistent with heating demand and reduced battery efficiency in cold conditions.

Table 2. Train/validation/test counts per vehicle and dataset group. Vehicle identifiers define cross-device clients, dataset groups define cross-silo and local clients, and the final row corresponds to centralized learning.

Identifier	**Train**	**Val**	**Test**	**Total**	
PostVan EV1	207	44	45	296	Cross-device FL
PostVan EV2	195	42	42	279	
PostVan EV3	218	47	47	312	
PostVan EV4	209	45	45	299	
PostVan EV5	204	44	43	291	
PostVan EV6	190	41	40	271	
PostVan EV7	197	42	43	282	
PostVan EV8	173	37	37	247	
PostVan EV9	203	44	43	290	
PostVan EV10	208	45	44	297	
BEV-D1	86	18	19	123	
BEV-D2	179	38	39	256	
ESOGU	2	1	1	4	
Postal	2004	431	429	2864	Cross-silo FL & local
Dynamic BEV	265	56	58	379	
ESOGU	2	1	1	4	
all	2271	488	488	3247	Centralized learning

4.2 Learning Settings

All models predict three quantiles of the daily energy demand distribution, $\tau \in \{0.1, 0.5, 0.9\}$, corresponding to lower, median, and upper demand estimates. This formulation is suitable for planning tasks where uncertainty awareness is important. The median represents the typical expected demand, while the upper quantile captures high-demand scenarios relevant to capacity reservation. Training minimizes the summed pinball loss across quantiles:

$$\mathcal{L}(w) = \frac{1}{|\mathcal{B}|} \sum_{i \in \mathcal{B}} \sum_{j=1}^{3} \rho_{\tau_j}\left(y_i - \hat{y}_{\tau_j,i}\right), \tag{9}$$

where $\rho_\tau(u) = u(\tau - \mathbf{1}[u < 0])$ and $\mathcal{B}$ denotes a mini-batch.

We evaluate three neural tabular architectures.

MLP. A shared trunk with two hidden layers of sizes $(128, 64)$, ReLU activations, batch normalization, and dropout ($p = 0.15$). Separate quantile heads of hidden dimension 32 are applied after the shared representation.

TabNet. Two attentive feature-selection steps, each computing a soft feature mask and processing the masked input through a shared feature block of hidden size 128. A complementary prior discourages repeated feature selection across steps. Step outputs are summed and passed to a linear quantile head.

Transformer. Each input feature is treated as a token, projected to a hidden dimension of 128 and combined with learnable feature embeddings. A Trans-

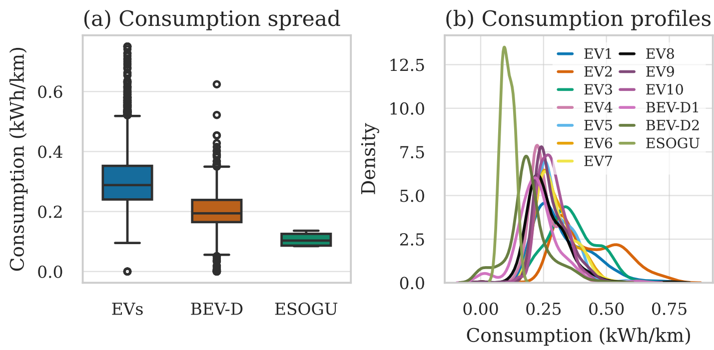


Fig. 3. Consumption distribution. Panel (a) compares the consumption spread across dataset groups. Panel (b) shows per-vehicle consumption density profiles.

former encoder with 2 layers, 4 attention heads, and feedforward dimension 512 processes the tokens. Representations are mean-pooled before the quantile head.

All models are trained for up to 80 epochs, or 80 server aggregation rounds in FL, using Adam with mini-batch size 128 and early stopping with patience 8. The checkpoint with the lowest validation MAE is used for test evaluation. In centralized and local training, standard supervised optimization is used. In FL, each selected client trains locally for up to 3 epochs. For cross-device FL, 4 clients are sampled per round from the 13 vehicle clients, with buffer size $B = 4$. For cross-silo FL, all 3 dataset-group clients participate in each round, with buffer size $B = 3$. Server-side aggregation follows Algorithm 1.

Models are evaluated using Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE), reported in kWh. For the FL experiments, we additionally report orchestration-level metrics (e.g., communication ingress, egress, and latency) to assess practical deployability under heterogeneous client conditions.

4.3 Results

All results are means over five random seeds with standard deviations in parentheses. The full feature configuration, i.e., *base* (Eq. 6), vehicle embeddings (VE) and *weather* information is used throughout unless stated otherwise.

Centralized versus Federated Learning. Table 3 reports the test MAE and RMSE across all learning settings and architectures. Under centralized training, the Transformer achieves the lowest MAE and RMSE (2.433 kWh and 3.417 kWh, respectively), with TabNet close behind (2.444 kWh MAE, 3.471 kWh RMSE) and MLP trailing (2.695 kWh MAE, 3.764 kWh RMSE).

In the cross-device (CD) setting, the global model trained across the 13 vehicle clients degrades modestly relative to centralized, with TabNet reaching 2.585 kWh MAE (+0.141 kWh over centralized TabNet) and the Transformer reaching

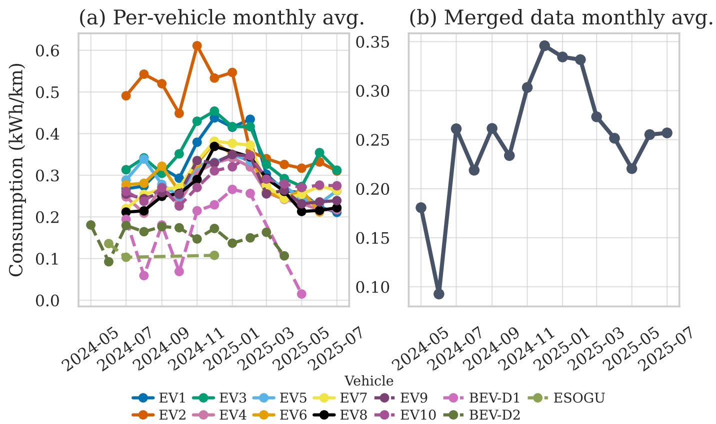


Fig. 4. Monthly average daily consumption. Panel (a) reports per-vehicle monthly averages. Panel (b) reports the merged monthly average across all vehicle-day samples.

Table 3. Test MAE and RMSE (kWh) for the full feature set across learning settings and architectures. Results are means over five seeds (std in parentheses). CD = cross-device; CS = cross-silo. **Bold** marks the best result per metric column.

	Centralized		CD Global		CD FT		CS Global		CS FT	
Model	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**
MLP	2.695	3.764	3.069	4.010	2.345	3.247	4.098	5.155	2.894	3.890
(std)	(0.288)	(0.305)	(0.202)	(0.199)	(0.049)	(0.065)	(0.496)	(0.497)	(0.578)	(0.603)
TabNet	2.444	3.471	2.585	3.535	**2.124**	**2.967**	**2.027**	**3.088**	**1.839**	**2.924**
(std)	(0.133)	(0.271)	(0.030)	(0.040)	(0.038)	(0.054)	(0.052)	(0.055)	(0.102)	(0.102)
Transformer	**2.433**	**3.417**	2.698	3.635	2.538	3.507	2.844	3.994	2.033	3.373
(std)	(0.170)	(0.308)	(0.157)	(0.132)	(0.165)	(0.184)	(0.851)	(0.864)	(0.592)	(0.674)

2.698 kWh MAE (+0.265 kWh). Cross-device fine-tuning (CD FT), in which the global model is further fine-tuned on each client's local data after federated training, recovers this gap. TabNet CD FT achieves 2.124 kWh MAE and 2.967 kWh RMSE, which is 0.320 kWh below the centralized TabNet baseline, demonstrating that personalization through fine-tuning can improve upon the centralized model. MLP CD FT similarly improves to 2.345 kWh MAE, reducing the global CD error by 0.724 kWh. The Transformer CD FT, however, shows only modest improvement over its global counterpart (2.538 vs. 2.698 kWh MAE), suggesting that the higher-capacity architecture is less amenable to local adaptation, possibly due to the limited amount of client-specific data available during fine-tuning. Transformer-based models typically require larger training corpora to fully exploit their representational capacity and to avoid unstable parameter adaptation under low-sample regimes, which is consistent with the relatively small number of vehicle-day samples available per client in the cross-device setting.

Table 4. Feature ablation under centralized training (all architectures), cross-device FT TabNet, and cross-silo FT TabNet. **Bold** marks the best MAE per setting.

	Cent. MLP		**Cent. TabNet**		**Cent. Transf.**		**CD FT TabNet**		**CS FT TabNet**	
Features	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**
EV only	2.898	4.015	2.806	3.889	2.955	4.099	2.569	3.797	3.685	4.510
EV + VE	2.677	3.701	2.485	3.382	2.772	3.822	2.483	3.684	2.378	3.251
EV + W	2.968	4.119	2.862	3.908	2.897	4.019	2.127	2.986	1.934	2.976
EV + VE + W	**2.695**	**3.764**	**2.444**	**3.471**	**2.433**	**3.417**	**2.124**	**2.967**	**1.839**	**2.924**

The cross-silo (CS) setting represents collaboration across three heterogeneous dataset groups with highly uneven sample sizes. TabNet is again the strongest model, with CS global TabNet reaching 2.027 kWh MAE, while CS,FT TabNet achieves the best aggregate result, with 1.839 kWh MAE and 2.924 kWh RMSE. This indicates that silo-level fine-tuning is beneficial in the evaluated setting, where each silo represents a distinct operational distribution. However, because the test set is dominated by postal-operation samples and ESOGU is extremely small, this result should not be interpreted as a general claim that cross-silo FL always outperforms centralized training. MLP and Transformer also benefit from fine-tuning, although the Transformer shows higher variance under cross-silo training.

Across all settings, TabNet is the strongest architecture as it achieves the best fine-tuned result in both FL partitioning schemes and is competitive with the Transformer under centralized training despite being a lighter model.

Feature Ablation. Table 4 reports the effect of progressively adding feature groups to the base EV telemetry signal under centralized training and for the two best fine-tuned FL configurations (CD FT and CS FT TabNet). Starting from the base configuration, adding vehicle embeddings consistently reduces error across all model and setting combinations. For centralized TabNet, the addition of vehicle embeddings reduces MAE from 2.806 to 2.485 kWh (-0.321 kWh), and for centralized MLP from 2.898 to 2.677 kWh (-0.221 kWh). This improvement confirms that learned embeddings capture EV-specific consumption characteristics that are not explained by the shared input features alone.

Adding weather data to the base feature set without EV embeddings improves performance in the FL fine-tuned settings more markedly than under centralized training. For CD FT TabNet, it achieves 2.127 kWh MAE, which is already below the centralized TabNet baseline, while for CS FT TabNet it reaches 1.934 kWh MAE. By contrast, under centralized training, the weather data slightly degrades performance relative to the base features for MLP and Transformer.

The full feature set achieves the lowest MAE and RMSE in every model and setting. For centralized TabNet the full configuration reduces MAE from 2.806 to 2.444 kWh; for CS FT TabNet the reduction is from 3.685 to 1.839 kWh, a decrease of 1.846 kWh. The consistent superiority of the full feature set across architectures and learning regimes confirms that vehicle embeddings and

Table 5. Test MAE and RMSE (kWh) for the local setting using TabNet. Each model is trained on one dataset group and evaluated on all test groups. Results are means over five seeds (std in parentheses).

	Test on EVs		Test on BEV		Test on ESOGU		Overall	
Train on	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**	**MAE**	**RMSE**
BEV	8.835	10.181	**4.511**	8.515	6.444	6.444	7.986	9.637
(std)	(0.399)	(0.476)	(0.230)	(0.482)	(1.761)	(1.761)	(0.397)	(0.507)
ESOGU	34.842	38.103	19.593	28.328	**1.418**	**1.418**	29.925	33.778
(std)	(37.161)	(41.549)	(19.959)	(26.315)	(1.011)	(1.011)	(31.616)	(35.969)
EVs	**2.229**	**2.940**	6.280	**8.276**	8.679	8.679	**3.349**	**4.203**
(std)	(0.070)	(0.094)	(1.450)	(1.405)	(4.529)	(4.529)	(0.265)	(0.216)

weather descriptors are complementary, i.e., embeddings capture inter-vehicle heterogeneity while weather features account for environmental variability that affects both energy consumption and battery efficiency.

Local Training and Cross-Dataset Transfer. Table 5 evaluates TabNet models trained on a single dataset group and tested on all three groups, to assess how well locally trained models generalize and to provide a lower bound on collaborative learning. When trained on the postal EVs and tested within the same group, the local model achieves 2.229 kWh MAE and 2.940 kWh RMSE, the strongest within-group result of the three local models. However, generalizing this model to the Dynamic BEV vehicles degrades accuracy substantially (6.280 kWh MAE), and to ESOGU (8.679 kWh MAE), reflecting the large operating differences between commercial postal vans and the public benchmark vehicles.

Training locally on BEV yields 4.511 kWh MAE within group, but predicting on the postal EVs is highly inaccurate (8.835 kWh MAE), with the overall MAE being 7.986 kWh. The ESOGU-trained model is practically unusable outside its own group, with MAE of 34.842 kWh on the postal EVs and 19.593 kWh on Dynamic BEV. This is expected given that the ESOGU group contributes only 4 valid vehicle-day samples after preprocessing, making local training unreliable.

These results highlight the limitations of purely local training under data scarcity and cross-dataset shift. The best local model reaches 3.349 kWh overall MAE, compared with 1.839 kWh for CS,FT TabNet. Thus, FL with fine-tuning provides a meaningful aggregate gain in this heterogeneous setting. However, ESOGU has only one test sample, so its local and transfer metrics are diagnostic rather than statistically robust.

4.4 Raspberry Pi Deployment

To evaluate the practical feasibility of the proposed FL pipeline, we deployed the cross-silo setting on a four-node Raspberry Pi testbed. One device acts as the FL server and the remaining three act as clients corresponding to the three dataset groups. This setup is the closest to the intended real-world deployment, where each data owner keeps raw records locally and exchanges model parameters.

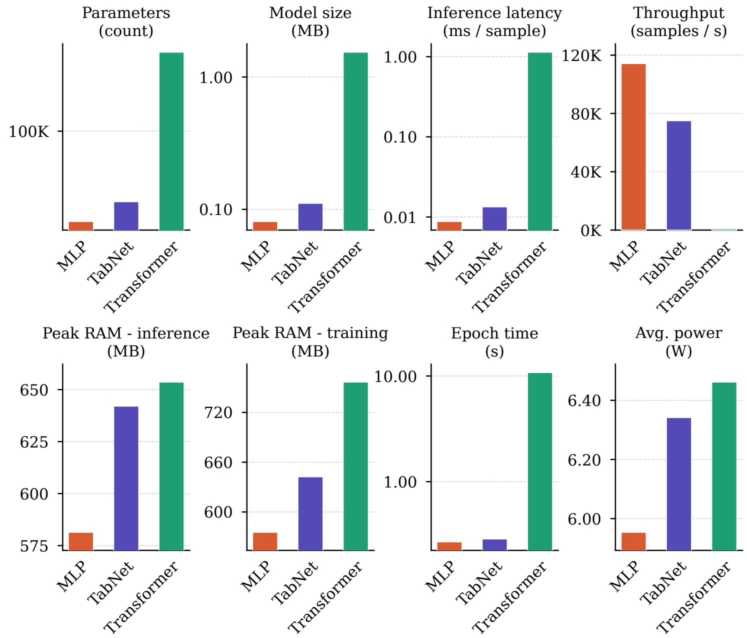


Fig. 5. Raspberry Pi profiling results for the three model architectures, measured on the client with the largest local dataset (private postal-operation silo). Metrics include inference latency, epoch time, throughput, peak RAM, and average power draw.

The deployed system follows the same buffered asynchronous protocol used in simulation. Each client polls the server, receives the current global model, performs local training, and submits the updated model together with local metrics. The server aggregates received updates when the buffer condition is satisfied. Predictive accuracy is computed from the same MAE and RMSE metrics used in simulation, and the Raspberry Pi execution produced the same ranking and no substantial accuracy degradation compared to simulation mode, confirming that deployment affects mainly the runtime and communication behavior.

Communication cost is measured directly based on the HTTP communication. For every request-response exchange, we record the request payload size, response payload size, and endpoint latency. Round-level ingress and egress are computed as the total bytes received and sent by the server per round, converted to MB. The communication overhead ratio is defined as:

$$r_{\text{comm}} = \frac{\text{total protocol bytes}}{\text{serialized model-state bytes}}, \tag{10}$$

where a value close to 1 indicates that communication volume is dominated by model parameters with limited additional protocol overhead.

Fig. 5 reports model-level profiling on the largest client, corresponding to the private postal-operation silo. The Transformer has the largest parameter count,

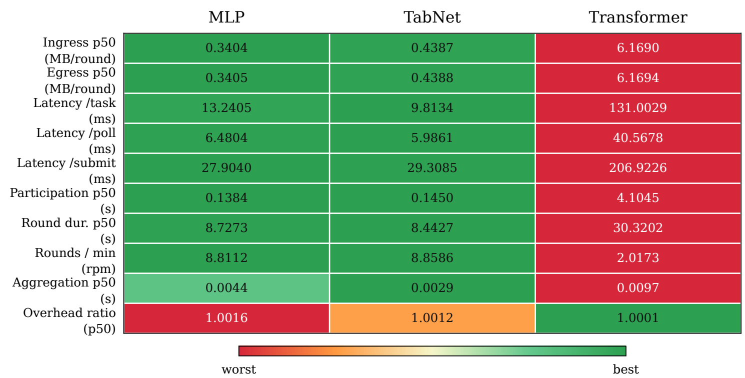


Fig. 6. Communication and orchestration metrics in the Raspberry Pi deployment. Green indicates better values and red indicates worse values.

which directly translates into higher inference latency, longer epoch time, and lower throughput. The MLP is the smallest and fastest model, with the lowest inference latency and highest throughput. TabNet lies between the two: more expensive than the MLP but substantially lighter than the Transformer. Peak RAM remains within budget for all architectures, although the Transformer requires the most memory during both inference and training. Average power differs only moderately across models, suggesting that the primary practical cost of larger architectures is runtime rather than instantaneous power draw.

Fig. 6 summarizes communication and orchestration metrics. The MLP and TabNet require less than 0.5 MB of ingress and egress per round, while the Transformer requires ≈ 6.17 MB in each direction, directly reflecting its larger serialized model state. Similarly, MLP and TabNet complete task assignment, submission, and aggregation with low latency, whereas the Transformer introduces substantially higher submission and aggregation delays, most visibly in participation time. MLP and TabNet complete several rounds per minute on the testbed, while the Transformer is considerably slower. The communication overhead ratio is close to 1 for all models, confirming that communication volume is dominated by model parameters rather than HTTP or metadata overhead.

Overall, the deployment confirms that buffered asynchronous FL is feasible on low-cost edge hardware. However, architecture choice has a strong effect on communication volume and round latency, with lightweight models offering an attractive balance between predictive quality and deployability.

5 Conclusion

In this paper, we presented a framework for daily EV energy demand prediction under centralized, cross-device, and cross-silo FL settings. A unified preprocessing pipeline transforms heterogeneous public and private EV records into vehicle-day samples, while the final model combines telemetry, vehicle embeddings, and weather descriptors. Results show that FL can approach centralized performance while preserving data locality, and the Raspberry Pi deployment confirms the feasibility of buffered asynchronous FL on resource-constrained hardware.

Future work will consider richer planning signals, simpler non-DL baselines, alternative FL algorithms, more balanced datasets, repeated splits, and sensitivity to noisy expected-distance estimates. Although FL preserves raw data locality, stronger privacy guarantees also require secure aggregation, differential privacy, and evaluation against inference or adversarial client attacks.

Acknowledgments. This work is supported by the European Commission under Grant No 101189589 through the Horizon EU program for the project "Open Cloud-EdgeIoT Platform Uptake in Large Scale Pilots (O-CEI)".

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Akshay, K., Grace, G.H., Gunasekaran, K., Samikannu, R.: Power consumption prediction for electric vehicle charging stations and forecasting income. Scientific reports **14**(1), 6497 (2024)
2. Amin, A., Amin, M.S., Park, H., Lee, D.: Electric vehicle range prediction models: A systematic review of machine learning, mathematical, and simulation approaches. World Electric Vehicle Journal **16**(11), 607 (2025). `https://doi.org/10.3390/wevj16110607`
3. Cavus, M., Ayan, H., Dissanayake, D., Sharma, A., Deb, S., Bell, M.C.: Forecasting electric vehicle charging demand in smart cities using hybrid deep learning of regional spatial behaviours. Energies **18**(13), 3425 (2025). `https://doi.org/10.3390/en18133425`
4. Chen, P., Qin, J., et al.: Electric vehicle charging demand forecasting at charging stations under climate influence for electricity dispatching. IET Power Electronics (2025). `https://doi.org/10.1049/pel2.12833`
5. van Etten, T., Degeler, V., Luo, D.: Large-scale forecasting of electric vehicle charging demand using global time series modeling. In: Proceedings of the 10th International Conference on Vehicle Technology and Intelligent Transport Systems (VEHITS). pp. 40–51 (2024). `https://doi.org/10.5220/0012555400003702`
6. Han, Q., Li, X.: A vertical federated learning method for electric vehicle charging station load prediction in coupled transportation and power distribution systems. Processes **13**(2), 468 (2025). `https://doi.org/10.3390/pr13020468`
7. Hussain, I., Ching, K.B., Uttraphan, C., Tay, K.G., Noor, A., Memon, S.A.: Optimizing electric vehicle energy consumption prediction through machine learning and ensemble approaches. Scientific Reports (2025), `https://www.nature.com/articles/s41598-025-14129-2`
8. Jin, R., Liu, H.: Privacy-aware electric vehicle load forecasting via blockchain-based federated transfer learning. Complex & Intelligent Systems (2025). `https://doi.org/10.1007/s40747-025-02002-8`
9. Marlin, R., Jurdak, R., Abuadbba, A.: H-fltn: A privacy-preserving hierarchical framework for electric vehicle spatio-temporal charge prediction (2025), `https://arxiv.org/abs/2502.18697`

10. Marzbani, F., Osman, A.H., Hassan, M.S.: Electric vehicle energy demand prediction techniques: An in-depth and critical systematic review. IEEE Access **11**, 96242–96255 (2023)
11. McMahan, B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data. In: Artificial intelligence and statistics. pp. 1273–1282. Pmlr (2017)
12. Narayana Gowda, S., et al.: Short-term aggregate electric vehicle charging load forecasting in diverse conditions with minimal data using transfer and meta-learning. Energy Systems (2024). https://doi.org/10.1007/s12667-024-00682-8
13. Nguyen, J., Malik, K., Zhan, H., Yousefpour, A., Rabbat, M., Malek, M., Huba, D.: Federated learning with buffered asynchronous aggregation. In: International conference on artificial intelligence and statistics. pp. 3581–3607. PMLR (2022)
14. Perifanis, V., Pavlidis, N., Yilmaz, S.F., Wilhelmi, F., Guerra, E., Miozzo, M., Efraimidis, P.S., Dini, P., Koutsiamanis, R.A.: Towards energy-aware federated traffic prediction for cellular networks. In: 2023 Eighth International Conference on Fog and Mobile Edge Computing (FMEC). pp. 93–100. IEEE (2023)
15. Perifanis, V., Nikolaidou, F., Pavlidis, N., Thomakos, P., Sendros, A.: Federated learning for early prediction of ev charging demand (2026), https://arxiv.org/abs/2605.04993
16. Polat, A.A., Keser, S.B., Sarıçiçek, İ., Erdoğmuş, A.K., Kafall, A., Yazıcı, A.: A dataset for state of charge and range estimation of an l5 type electric vehicle that is used for urban logistic. Data in Brief p. 111933 (2025)
17. Rachavelpula, S.V.R., Cha, S.: Personalized electric vehicle energy consumption estimation framework that integrates driver behavior with map data (2026), https://arxiv.org/abs/2604.20764
18. Szumska, E.M., et al.: Machine learning applications in energy consumption forecasting and management for electric vehicles: A systematic review. Energies **18**(20), 5420 (2025). https://doi.org/10.3390/en18205420
19. Tian, J., Liu, H., Gan, W., et al.: Short-term electric vehicle charging load forecasting based on tcn-lstm network with comprehensive similar day identification. Applied Energy **381**, 125174 (2025). https://doi.org/10.1016/j.apenergy.2024.125174
20. Tritsarolis, A., Sampaio, G., Pelekis, N., Theodoridis, Y.: On electric vehicle energy demand forecasting and the effect of federated learning. arXiv preprint arXiv:2602.20782 (2026)
21. Wang, Y., Huang, H., et al.: Prediction of battery electric vehicle energy consumption via pre-trained model under inconsistent feature spaces. Technologies **13**(11), 493 (2025). https://doi.org/10.3390/technologies13110493
22. Wen, J., Zhang, Z., Lan, Y., Cui, Z., Cai, J., Zhang, W.: A survey on federated learning: challenges and applications. International journal of machine learning and cybernetics **14**(2), 513–535 (2023)
23. Yasko, M., Moussa Issaka, A., Tian, F., Kazmi, H., Driesen, J., Martinez, W.: Unveiling energy dynamics of battery electric vehicle using high-resolution data. Scientific Data **12**(1), 1878 (2025)
24. Yin, W., Ji, J.: Prediction of ev charging load based on federated learning. Energy **316**, 134559 (2025). https://doi.org/10.1016/j.energy.2025.134559
25. Zhang, X., Zhang, Z., Liu, Y.: A review of machine learning approaches for electric vehicle energy consumption modelling in urban transportation. Renewable Energy **234**, 121243 (2024). https://doi.org/10.1016/j.renene.2024.121243