MatchEstimate: A Robust Aggregation Method for Federated Learning, Electric Vehciles Case Study

Mohamed Redha Mahamdi*, Ahmed-Ramzi Houalef†, Lydia Douaidi†, Florian Delavernhe†, Sidi-Mohammed Senouci†
Université Bourgogne Europe, DRIVE UR 1859, 58000 Nevers, France
*Email: redha.mahamdi@ube.fr
†Email: {firstname.lastname}@ube.fr

***Abstract*—The rise of Electric Vehicles (EVs) is driving demand for intelligent, data-driven solutions in mobility and energy management. Federated Learning (FL) enables collaborative model training across distributed EV systems while preserving data privacy. However, standard FL algorithms struggle with client heterogeneity—such as non-IID (non-Independent and non-Identically Distributed) data, varying compute power, and unstable communication. We propose MatchEstimate (ME), a robust, flexible aggregation method designed for heterogeneous FL. As a drop-in replacement for strategies like Federated Averaging (FedAvg), ME extends our prior method, FedEstimate, to improve performance and training stability across diverse clients. In an EVs case study, ME significantly enhances their accuracy under realistic conditions, advancing privacy-preserving, decentralized intelligence for next-gen mobility.**

***Index Terms*—Federated Learning, Aggregation Method, Client Heterogeneity, Electric Vehicles, Edge Intelligence, Privacy-Preserving Machine Learning, Non-IID Data, Distributed Learning, Smart Mobility**

I. Introduction

The growing adoption of Electric Vehicles (EVs) is transforming the transportation and energy sectors, creating new opportunities for intelligent, data-driven applications. Modern EVs generate a rich stream of data related to energy consumption, driving behavior, battery health, environmental conditions, and user preferences. Leveraging this data can significantly enhance key services such as battery management, route optimization, and predictive maintenance. One of the most pressing concerns for EV users is *range anxiety*—the fear that a vehicle will not have sufficient battery charge to reach its destination. Accurately *predicting energy consumption* under varying conditions is essential for alleviating this concern and improving the reliability of EV systems. However, collecting and centralizing this data for analysis or model training is often infeasible due to privacy concerns, data ownership, and communication limitations. *Federated Learning (FL)* [1] offers a compelling solution by enabling the collaborative training of machine learning models across a network of distributed EVs without the need to share raw data. FL preserves privacy and reduces data transfer costs, making it ideal for EV fleets and large-scale mobility infrastructures. Despite its promise, FL faces major challenges in real-world deployments—chief among them is *heterogeneity*. In EV networks, heterogeneity is ubiquitous: vehicles differ in usage patterns, geographic environments, hardware specifications, and driver behavior. This leads to *statistical heterogeneity* (non-IID i.e., non-Independent and non-Identically Distributed data, where each client holds data drawn from different distributions), *system heterogeneity* (varying device and network capabilities), and *participation heterogeneity* (irregular client availability). These forms of variability can severely impact the stability and performance of FL algorithms, with standard aggregation methods like *Federated Averaging (FedAvg)* [1] often failing to converge or producing biased models. In this paper, we propose **MatchEstimate (ME)**, a novel aggregation method specifically designed to address heterogeneity in FL systems. ME is *a standalone FL algorithm*, and it is a promising *plug-and-play aggregation strategy* that can serve as a drop-in replacement for FedAvg in any FL framework. By enhancing the robustness of FL under heterogeneity, ME enables its deployment across diverse real-world scenarios. In this paper, we focus on a key application, energy consumption prediction for EVs to demonstrate its effectiveness. Improving model reliability in this context contributes to reducing range anxiety and promoting smarter, more efficient EV usage. The main contributions of this paper are:

- Introduces ME, a new learned aggregation operator that combines neuron matching with data-driven estimation, offering a more robust and flexible alternative to traditional aggregation methods.
- Various state-of-the-art federated aggregation techniques and personalisation strategies are tested with and without ME. Incorporating ME consistently strengthens these approaches, delivering performance gains for methods like FedPer, FedRep, and Ditto.
- Provides a curated and physics-informed from realistic dataset combining two public driving datasets for four Nissan Leaf vehicles, creating a realistic benchmark for energy consumption prediction under non-IID conditions.

The remainder of this paper is organized as follows. Section II reviews existing work related to energy modeling and prediction for EVs, as well as methods addressing heterogeneity in FL. Section III introduces the proposed ME method, outlining its core intuition, design principles, and implementation. Section IV presents experimental evaluations across multiple benchmarks under various heterogeneity scenarios, demonstrating the effectiveness of the approach. Finally, Sec-

tion V summarizes the main findings and discusses potential directions for future work.

II. Related Work

In this section, we first cover standard FL baselines and personalized FL (PFL) methods, then describe FedEstimate [2] separately as it is key to understanding our proposed MatchEstimate algorithm. We conclude with an overview of research on range anxiety and how FL can help address it.

A. FL Methods for Addressing Client Heterogeneity

FL [1] was initially proposed with the goal of enabling collaborative model training across decentralized clients without sharing raw data. One of the most prominent baseline methods is **FedAvg** [1], which averages locally trained models to create a global model. While FedAvg is simple and effective in IID settings, it often struggles in non-IID environments due to client drift and poor convergence. To address this, **FedProx** [3] introduces a proximal term in the local objective function to limit drastic updates and reduce divergence from the global model. **FedNova** [4] (Normalized Objective Value Aggregation) enhances robustness by normalizing client contributions during aggregation, thus reducing the impact of skewed local updates. Another improvement, **SCAFFOLD** [5], tackles the issue of client drift directly by using control variates to correct the direction of local updates and align them with the global optimization path. Other strategies to handle heterogeneity is finding alignments in neural networks such as **FedMA** [6] align hidden layer neurons across client models using layer-wise matching, allowing each client to maintain a personalized model structure while benefiting from collaborative training. More recently, **MOON** [7] introduces a contrastive learning approach to enhance consistency between local and global representations, reducing representation drift in non-IID scenarios. These methods primarily focus on improving convergence and stability during training, without altering the model architecture or introducing personalization. As such, they are often used as baselines or optimization-enhancing techniques in general FL setups.

In practical FL applications, client data is rarely identically distributed, leading to degraded model performance when using global models alone. To handle such statistical heterogeneity, personalization-oriented methods have been developed. **Ditto** [8] proposes a bi-level optimization framework where each client learns both a local personalized model and a shared global model, enabling better generalization under heterogeneity. Similarly, **FedPer** [9] addresses personalization by splitting the model into shared base layers and client-specific output layers, allowing clients to adapt to local tasks while leveraging shared knowledge. In contrast, **FedRep** [10] reverses this architecture by personalizing the feature extractor while keeping the classifier shared—beneficial when client-level representations differ substantially. These methods represent a significant evolution in FL, moving from a one-size-fits-all global model toward adaptable and client-aware learning strategies.

Most FL aggregation methods, such as FedProx and FedPer, rely on simple data-size–weighted averaging of client parameters. While effective in certain cases, this averaging approach often fails to capture the complex, client-specific variations present in non-IID data. As a result, it can lead to sub-optimal global models and slower convergence, and degrading accuracy. To address these challenges, our prior work introduced FedEstimate [2], a patented aggregation framework, which replaces the fixed averaging rule with a MultiLayer Perceptron (MLP) deployed on the server. Instead of explicitly averaging client parameters, the MLP learns to predict the next global model based on the set of received client updates. This adaptive, data-driven strategy enables the server to better capture the complex relationships among heterogeneous clients, providing greater flexibility than traditional rule-based aggregation methods.

Figure 1 outlines the steps of FedEstimate executed in each communication round, as follows:

1) **Model broadcast.** The server initialises (or re-initialises) a base model $M_0^{(t)}$ and sends it to all n clients.
2) **Client-side training.** Each client e_i:
   a) splits its local dataset into n non-IID shards $D_{i1}, \ldots, D_{in}$;
   b) trains $n+1$ **models** with identical architecture
   — one on the full data D_i to obtain weights W_i;
   — one on each shard D_{ij} to obtain weights W_{ij};
   c) sends the full collection $\{W_i, W_{i1}, \ldots, W_{in}\}$ back to the server.
3) **Server-side "weight dataset" construction.** For every parameter index k the server forms a feature vector $\mathbf{x}_k = [\,W_{11}[k], \ldots, W_{1n}[k], \ldots, W_{n1}[k], \ldots, W_{nn}[k]\,]^{\mathsf{T}}$ and a target scalar $y_k = \frac{1}{n}\sum_{i=1}^{n} W_i[k]$.
4) **Learning the aggregator.** An MLP regressor g_θ is trained on the dataset $\{(\mathbf{x}_k, y_k)\}_k$; the loss is mean-squared error.
5) **Predicting the global model.** At inference time the MLP consumes the *current round's* concatenated shard weights (features only) and outputs a *predicted* weight vector $\widehat{W}^{(t)} = g_{\theta^{(t)}}(\mathbf{x}^{(t)})$, which becomes the new global model.
6) **Repeat.** The server broadcasts $\widehat{W}^{(t)}$ to all clients and the process repeats until convergence.

FedEstimate therefore treats aggregation as a data-driven regression problem, enabling the server to discover a sophisticated, round-adaptive combination of client updates that is better suited to heterogeneous data than a fixed arithmetic mean. In this paper, we propose ME, an enhancement of FedEstimate, which incorporates a technique for aligning semantically equivalent neurons. This alignment ensures that weight-space operations, such as averaging or interpolation, become meaningful and consistent across clients.

B. Range anxiety leverage using FL

Several works treated range anxiety problem for EVs such as [11]. Presents an extended FL (E-FL) framework designed

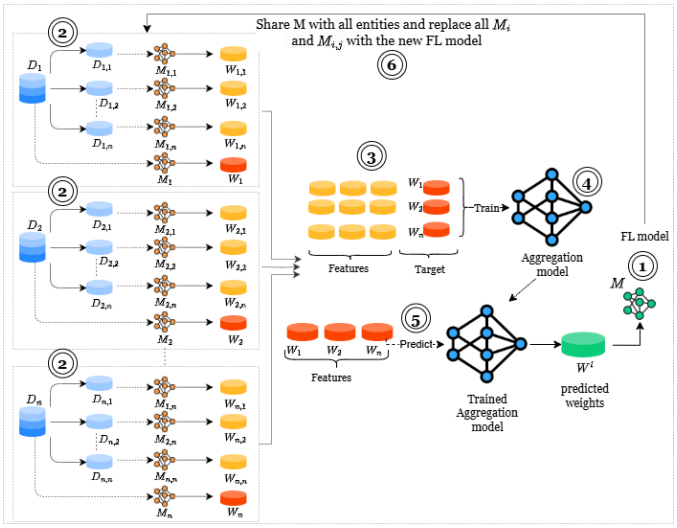

Fig. 1. Architecture overview of the FedEstimate approach.

to accurately estimate energy consumption for EVs while preserving driver privacy. Integrating local anomaly detection and a similarity-based sharing policy, the approach ensures robust learning even when vehicles exhibit diverse driving behaviors or generate abnormal data. However, the work relies on a basic linear model which may limit its capacity to capture complex driving dynamics, does not benchmark against standard FL algorithms, and does not explore how sensitive its performance is to manually set hyperparameters like similarity thresholds and anomaly detection settings. [12] proposes Fed-BEV, a FL framework that models Battery Electric Vehicles (BEVs) energy consumption using local stacked-LSTM predictors and the FedAvg algorithm to aggregate models without sharing raw data. It integrates realistic driving scenarios simulated in SUMO and energy evaluation via a Matlab/Simulink BEV model to generate training data. Similarly, [13] presents a framework for predicting energy consumption in BEVs with FL techniques to ensure user data privacy. The authors experiment with various machine learning models, including Random Forest, XGBoost, GRU, ANN, and LSTM, and identify LSTM as the most effective local model. They also evaluate five FL algorithms: FedSGD, FedAvg, FedProx, FedPer, and FedRep [1], [3], [9], [10] and determine that FedAvg achieves the best trade-off between accuracy and computational cost. Extensive experiments were conducted under various configurations to evaluate model robustness and deployment feasibility. The study concludes that FL when combined with appropriate models, can significantly enhance BEV energy consumption prediction accuracy while preserving user privacy. However, both papers present several limitations. Although they acknowledge the heterogeneity of vehicle data, they primarily rely on FedAvg [1] without thoroughly investigating more suitable FL methods designed to handle heterogeneity. Furthermore, for route planning scenarios, using LSTM as a local model is suboptimal. LSTMs alone often struggle with capturing long-range dependencies. Since LSTMs process information strictly as a linear sequence, they are ill-suited to address energy-aware routing, which is inherently a branching graph problem. At each intersection, a route planner must evaluate multiple possible outgoing roads or decide to insert a charging stop. However, an LSTM's hidden state does not explicitly represent these alternative paths and therefore cannot natively compare or select among them. Overall, while the papers offers valuable insights into privacy-preserving modeling for EVs, its deployment feasibility and methodological rigor could be significantly improved.

III. Problem Formulation

This section introduces ME, which aligns client models through neuron matching and learnable client weights aggregation. It also describes pre-processing and energy modeling used for prediction.

A. MatchEstimate

The proposed method combines FedMA's neuron permutation alignment with FedEstimate's aggregation strategy. It aligns neurons to make weight operations meaningful, captures intra-client heterogeneity by training multiple shard-specific models per client, and uses a learned aggregator at the server to predict full-client updates.

Let C denote the number of clients, where each client i owns private dataset D_i of size $n_i = |D_i|$. All clients share one neural architecture $g_{\boldsymbol{w}} \colon \mathbb{R}^{d_{\text{in}}} \to \mathbb{R}^{d_{\text{out}}}$ with parameter vector $\boldsymbol{w} \in \mathbb{R}^m$ (m is the number of parameters). Training proceeds in synchronous rounds $r = 0, \ldots, R-1$. At the start of every round the server broadcasts the current *baseline* weights $\boldsymbol{b}^{(r)}$.

Each client then performs three independent local trainings initialized with $\boldsymbol{b}^{(r)}$, as detailed in FedEstimate (see Section II-A).

In FL aggregation methods, the server averages corresponding weights across clients. However, neural networks are permutation-invariant—neurons with similar roles may be ordered differently, causing naive averaging to create a "Franken-model" with poor accuracy. FedMA [6] tackles this by aligning neurons before averaging, underscoring the need for proper neuron matching in federated aggregation. FedMA concatenates weights and biases into row vectors $\boldsymbol{v}_{i,j}$ for client i and $\boldsymbol{u}_j$ as the current reference vector. The similarity between neuron p of client i and neuron q of the reference is measured via the cosine distance:

$$\text{Cost}_{pq} = 1 - \frac{\langle \mathbf{v}_{i,p}, \mathbf{u}_q \rangle}{\|\mathbf{v}_{i,p}\|_2 \, \|\mathbf{u}_q\|_2} \tag{1}$$

The Hungarian algorithm takes as input the cosine distance matrix defined in Equation 1 to obtain the permutation $\pi \in S_{d_{\text{out}}}$ with minimal total cost. π is applied to the rows of the current layer and the columns of the next layer, preserving forward activations. The reference weights are updated as a running mean of all aligned clients.

After alignment, the shard deltas are stacked column-wise to construct the matrix :

$$\mathbf{C}_i = [\, \delta_{i1} \;\; \ldots \;\; \delta_{iK} \,] \in \mathbb{R}^{m \times K}. \tag{2}$$

For every parameter index j we form a feature–target pair:

$$\boldsymbol{x}_{ij} = (\delta_{i1}[j], \ldots, \delta_{iK}[j]) \in \mathbb{R}^K, \qquad y_{ij} = \Delta_i[j] \in \mathbb{R}, \quad (3)$$

where:

$$\delta_{ik} = \boldsymbol{w}_{ik} - \boldsymbol{b}^{(r)}, \qquad \Delta_i = \boldsymbol{w}_i^{\text{full}} - \boldsymbol{b}^{(r)}, \quad (4)$$

where $\boldsymbol{w}_{ik}$ represents the model trained on the k-th local shard of client i, and $\boldsymbol{w}_i^{\text{full}}$ denotes the personalized model obtained by client i through fine-tuning the global baseline $\boldsymbol{b}^{(r)}$.

Collecting all clients gives $\mathcal{D} = \{(\boldsymbol{x}_{ij}, y_{ij})\}_{i,j}$ with $C \times m$ rows. Each feature dimension is standardised; the targets are centred and scaled by a scalar factor. An MLP $f_{\boldsymbol{\theta}} \colon \mathbb{R}^K \to \mathbb{R}$ is trained on $\mathcal{D}$ to minimise the mean-square error :

$$\mathcal{L}(\boldsymbol{\theta}) \;=\; \frac{1}{C \times m} \sum_{i,j} \big(f_{\boldsymbol{\theta}}(\hat{\boldsymbol{x}}_{ij}) - \hat{y}_{ij}\big)^2. \quad (5)$$

To compute the client-specific update, each client i's predicted full-model delta $\tilde{\Delta}_i$ is obtained by:

$$\tilde{\Delta}_i = \text{denorm}\Big(f_{\boldsymbol{\theta}}\big((\mathbf{C}_i^\top - \boldsymbol{\mu})/\boldsymbol{\sigma}\big)\Big) \in \mathbb{R}^m. \quad (6)$$

Each client is assigned a weight proportional to its dataset size: $\alpha_i = \dfrac{n_i}{\sum_j n_j}$. The global model update is then computed as the weighted sum of the predicted deltas:

$$\Delta_{\text{global}}^{(r)} = \sum_{i=1}^{C} \alpha_i \, \tilde{\Delta}_i, \quad (7)$$

$$\boldsymbol{b}^{(r+1)} = \boldsymbol{b}^{(r)} + \Delta_{\text{global}}^{(r)}. \quad (8)$$

The proposed MatchEstimate method is summarized in Algorithm 1.

Algorithm 1 MatchEstimate

Require: Baseline $\boldsymbol{b}^{(0)}$, rounds R, shard count K

for $r \leftarrow 0$ **to** $R-1$ **do**
  broadcast $\boldsymbol{b}^{(r)}$ to all clients
  receive $(\Delta_i, \mathbf{C}_i)_{i=1}^{C}$
  build dataset $\mathcal{D}$
  train aggregator $f_{\boldsymbol{\theta}}$ with Eq. (5)
  for each clients i **do**
    $\tilde{\Delta}_i \leftarrow$ Eq. (6)
  end for
  $\Delta_{\text{global}}^{(r)} \leftarrow$ Eq. (7)
  $\boldsymbol{b}^{(r+1)} \leftarrow$ Eq. (8)
end for

B. Pre-processing and energy modeling

In the pre-processing stage, two raw Datasets are merged and harmonized to produce data for four BEV of Nissan leaf model [14]–[16], which serve as clients in a FL experiment. The dataset shows heterogeneity in both data size—ranging from about 3,000 to over 10,000 instances per client—and in data distribution, with distinct temperature and speed patterns as seen in Figure 2. The data undergoes several cleaning steps: duplicate rows are removed, sporadic gaps are interpolated, and columns with more than 90 % missing values are dropped. Since one source had 1Hz frequency, the remaining vehicles are down-sampled to a uniform 1 Hz. Units are then standardized (e.g., speed-limit ranges converted to single floats), and timestamps are reconstructed. A physics-based feature block is engineered at this point [17].

four force components are computed from our dataset:

$$F_{\text{roll}} = C_{rr}\, m\, g \quad \text{(rolling resistance)} \quad (9)$$

$$F_{\text{climb}} = m\, g\, \sin\theta \;\approx\; m\, g\, \text{grade} \quad \text{(climb force)} \quad (10)$$

$$F_{\text{aero}} = \tfrac{1}{2}\, \rho\, C_d\, A\, v^2 \quad \text{(aerodynamic drag)} \quad (11)$$

$$F_{\text{acc}} = m\, a \quad \text{(inertial force)} \quad (12)$$

These forces are computed using the following parameters: curb mass $m = 1680\,\text{kg}$,

rolling resistance coefficient $C_{rr} = 0.009$, drag coefficient $C_d = 0.28$, frontal area $A = 2.3\,\text{m}^2$, air density $\rho = 1.225\,\text{kg}\,\text{m}^{-3}$, speed v, acceleration a, and road grade θ, the latter derived from successive GPS elevations. These four force components, along with the per-second ambient temperature, form the five input features used to predict the vehicle's energy consumption.

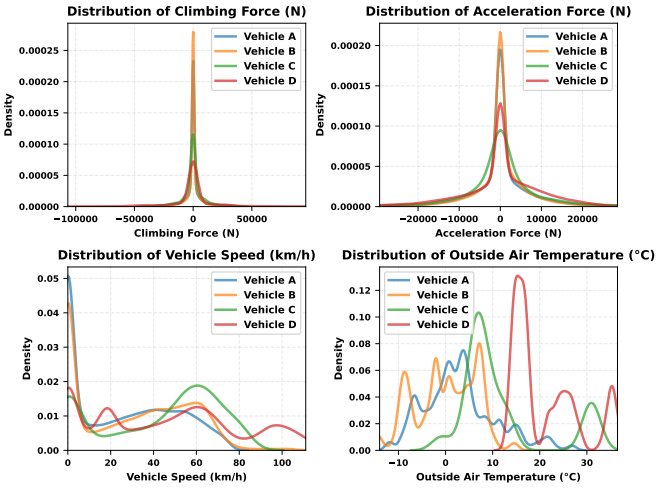

Fig. 2. Data distribution comparison across the four vehicles.

IV. Results

We used four vehicle datasets to evaluate different approaches. A global model trained on all data served as an upper bound, while local baselines were built by training one model per vehicle. For FL, we tested six standard algorithms—FedAvg, FedEstimate, FedProx, FedMA, FedNova, and MOON and created variants by replacing their FedAvg aggregation with our ME operator. Additionally, we evaluated

three PFL methods (FedPer, FedRep, Ditto) in both their vanilla form and with ME-enhanced aggregation.

Figure 3 shows the overall cross-evaluation, and Figure 4 compares the vanilla and ME-enhanced personalised methods.

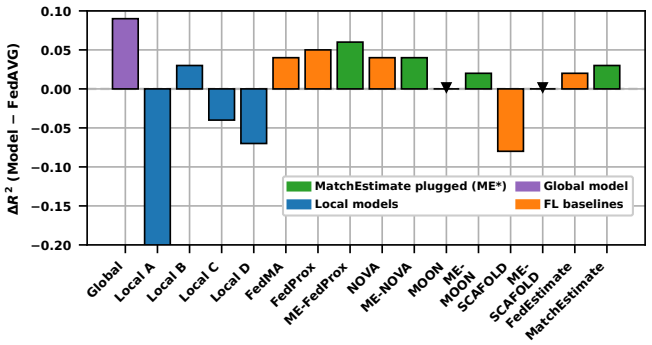

Fig. 3. Overall R^2: difference between each model and the FEDAVG baseline.

According to Figure 3, ME never degrades performance; in no experiment did ME fall below the FedAvg baseline. It consistently lifts every baseline it touches: for example, SCAFFOLD upgraded to ME-SCAFFOLD removes the negative dip and climbs back to FedAvg parity; FedProx upgraded to ME-FedProx adds approximately +0.01–0.02 ΔR^2; and MOON upgraded to ME-MOON adds about +0.01 ΔR^2.

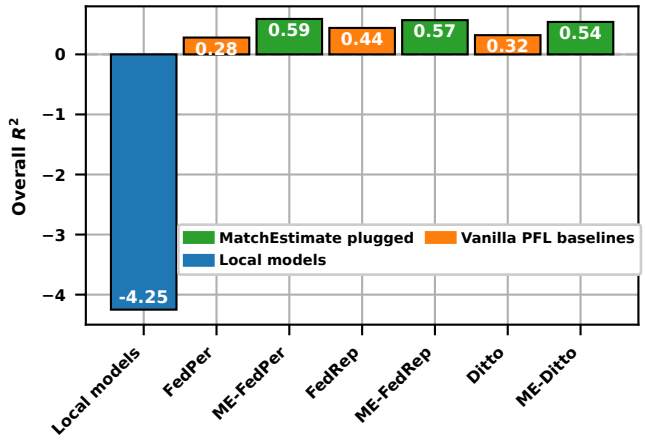

Fig. 4. Overall R^2 for each PFL method.

From Figure 4 and Figure 5, local, non-federated models form the lower bound with an overall score of -4.25 because they collapse off-client. Vanilla personalised-FL baselines such as FedPer, FedRep, and Ditto achieve only modest overall scores between 0.28 and 0.44. Injecting ME into the aggregation step lifts all these personalised baselines: FedPer upgraded to ME-FedPer shows the largest improvement with a +0.31 absolute R^2 increase, while FedRep and Ditto upgraded to ME variants gain between +0.13 and +0.22 absolute R^2. A notable plug-and-play benefit is that swapping to ME yields 0.1–0.3 absolute R^2 improvements without requiring a hyperparameter sweep.

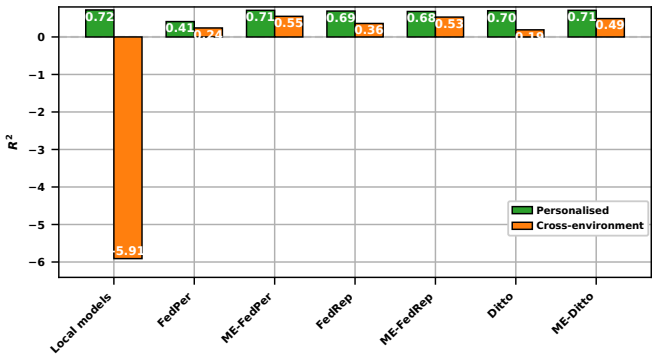

Fig. 5. Average R^2 obtained **on-device** (green) versus **cross-device** (orange) for every PFL variant.

As observed in the previous figures, the centralized model performed very well due to its access to the complete dataset, whereas the local models performed poorly because of their data heterogeneity. In contrast, FL models—particularly the ME variants—achieved intermediate performance, offering the best trade-off between the privacy protection of local models and the high accuracy of the centralized approach, thereby helping to mitigate range anxiety. By treating client weights as learnable entities within the aggregation process and aligning neurons across clients, ME potentially leads to improved generalization on non-IID datasets and better consistency in merging heterogeneous local models.

V. Conclusion

This paper introduces MatchEstimate, a novel FL aggregation technique designed to address the challenges of data heterogeneity (non-IID), with a specific application to energy consumption prediction for EVs. Experimental results on a real-world dataset demonstrate that ME outperforms state-of-the-art FL models, validating its effectiveness in heterogeneous environments.

In conclusion, ME proves to be a powerful and reliable addition for boosting the performance of various FL approaches, especially PFL. With further architectural optimizations and dedicated hyperparameter tuning, even larger improvements can be expected. ME's plug-and-play nature, demonstrates its versatility as a general-purpose enhancement for FL models.

Acknowledgment

This paper is supported by the OPEVA project that has received funding within the Chips Joint Undertaking (Chips JU) from the European Union's Horizon Europe Program and the National Authorities (France, Czechia, Italy, Portugal, Turkey, Switzerland), under grant agreement 101097267. Views and opinions expressed are however those of the author(s) only and do not necessarily reflect those of the European Union or Chips JU. Neither the European Union nor the granting authority can be held responsible for them.

References

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," 2023. [Online]. Available: https://arxiv.org/abs/1602.05629

[2] S.-M. S. Ahmed Saadallah, Lydia Douaidi, “Fedestimate: Federated learning approach with an aggregation model,” 2024, patent pending, Application ID: EP24305850.0, filed 29 march 2024.
[3] T. Li, A. K. Sahu, M. Zaheer, M. Sanjabi, A. Talwalkar, and V. Smith, “Federated optimization in heterogeneous networks,” 2020. [Online]. Available: https://arxiv.org/abs/1812.06127
[4] J. Wang, Q. Liu, H. Liang, G. Joshi, and H. V. Poor, “Tackling the objective inconsistency problem in heterogeneous federated optimization,” 2020. [Online]. Available: https://arxiv.org/abs/2007.07481
[5] S. P. Karimireddy, S. Kale, M. Mohri, S. J. Reddi, S. U. Stich, and A. T. Suresh, “Scaffold: Stochastic controlled averaging for federated learning,” 2021. [Online]. Available: https://arxiv.org/abs/1910.06378
[6] H. Wang, M. Yurochkin, Y. Sun, D. Papailiopoulos, and Y. Khazaeni, “Federated learning with matched averaging,” 2020. [Online]. Available: https://arxiv.org/abs/2002.06440
[7] Q. Li, B. He, and D. Song, “Model-contrastive federated learning,” 2021. [Online]. Available: https://arxiv.org/abs/2103.16257
[8] T. Li, S. Hu, A. Beirami, and V. Smith, “Ditto: Fair and robust federated learning through personalization,” 2021. [Online]. Available: https://arxiv.org/abs/2012.04221
[9] M. G. Arivazhagan, V. Aggarwal, A. K. Singh, and S. Choudhary, “Federated learning with personalization layers,” 2019. [Online]. Available: https://arxiv.org/abs/1912.00818
[10] L. Collins, H. Hassani, A. Mokhtari, and S. Shakkottai, “Exploiting shared representations for personalized federated learning,” 2023. [Online]. Available: https://arxiv.org/abs/2102.07078
[11] S. Zhang, “An energy consumption model for electrical vehicle networks via extended federated-learning,” in *2021 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, Jul. 2021, p. 354–361. [Online]. Available: http://dx.doi.org/10.1109/IV48863.2021.9575223
[12] M. Liu, “Fed-bev: A federated learning framework for modelling energy consumption of battery electric vehicles,” 2021. [Online]. Available: https://arxiv.org/abs/2108.04036
[13] S. Yan, H. Fang, J. Li, T. Ward, N. O’Connor, and M. Liu, “Privacy-aware energy consumption modeling of connected battery electric vehicles using federated learning,” *IEEE Transactions on Transportation Electrification*, vol. 10, no. 3, p. 6663–6675, Sep. 2024. [Online]. Available: http://dx.doi.org/10.1109/TTE.2023.3343106
[14] G. Wu, F. Ye, P. Hao, D. Esaid, K. Boriboonsomsin, and M. J. Barth, “Deep learning-based eco-driving system for battery electric vehicles,” 2019, dataset. [Online]. Available: https://doi.org/10.6086/D1FW9G
[15] S. Zhang, D. Fatih, F. Abdulqadir, T. Schwarz, and X. Ma, “Extended vehicle energy dataset (eved): an enhanced large-scale dataset for deep learning on vehicle trip energy consumption,” 2022. [Online]. Available: https://arxiv.org/abs/2203.08630
[16] S. Zhang, “Extended vehicle energy dataset (eved),” https://bitbucket.org/datarepo/eved-dataset, 2022, apache 2.0 licence. Accessed 12 June 2025.
[17] A.-R. Houalef, F. Delavernhe, S.-M. Senouci, and E. hassane Aglzim, “Utilizing data-driven techniques to improve predictive modeling of connected electric vehicle energy consumption,” in *Proceedings of the IEEE Vehicular Technology Conference (VTC-Fall)*, 2024.