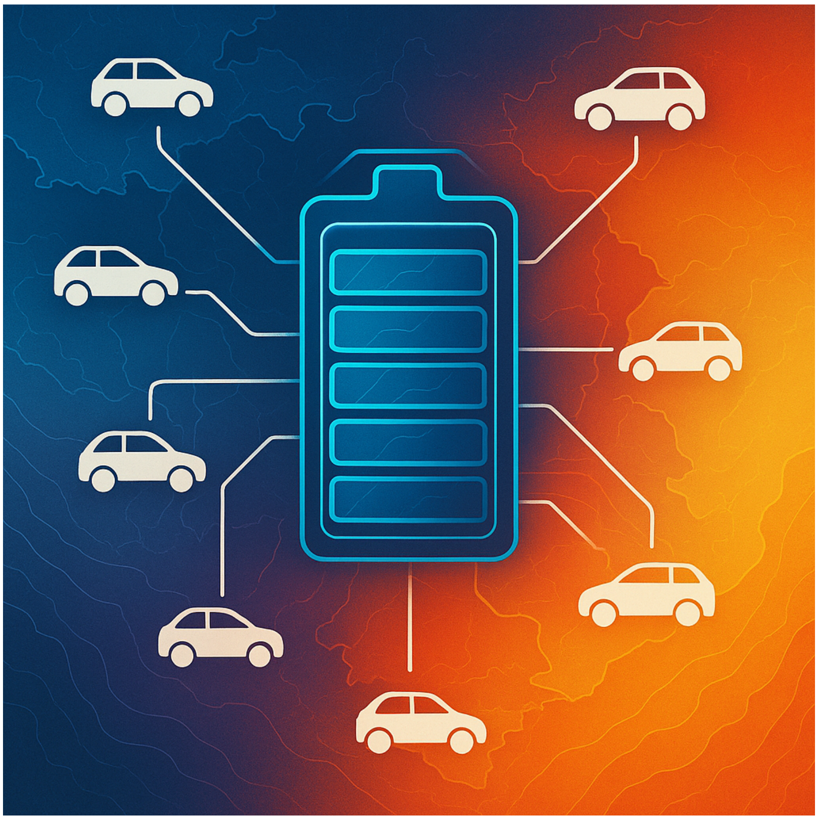

Federated Clustering of Electric Vehicle's Usage Patterns for Personalized State-of-Health Estimation

An Adaptive Learning Approach Using Dynamic Clustering

Master's thesis in Complex Adaptive Systems

AXEL LUNDKVIST

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2025
www.chalmers.se

Master's thesis 2025

Federated Clustering of Electric Vehicle's Usage Patterns for Personalized State-of-Health Estimation

An Adaptive Learning Approach Using Dynamic Clustering

AXEL LUNDKVIST

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Electrical Engineering
Division of Automatic Control
Chalmers University of Technology
Gothenburg, Sweden 2025

Federated Clustering of Electric Vehicle's Usage Patterns for Personalized State-of-Health Estimation
An Adaptive Learning Approach Using Dynamic Clustering
AXEL LUNDKVIST

Supervisor: Christian Fleischer, Cognivity AI AB
Examiner: Changfu Zou, Systems and Control, Electrical Engineering

Master's Thesis 2025
Department of Electrical Engineering
Division of Systems and Control Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2025

Federated Clustering of Electric Vehicle's Usage Patterns for Personalized State-of-Health Estimation
An Adaptive Learning Approach Using Dynamic Clustering
AXEL LUNDKVIST
Department of Electrical Engineering
Chalmers University of Technology

Abstract

Battery-health degradation varies significantly among electric vehicles, complicating accurate fleet-wide monitoring. This study investigates key factors influencing battery ageing and incorporates them into a dynamic clustered federated-learning framework, combining K-means initialization and Affinity Propagation (AP) clustering. Evaluated on NASA laboratory cells and telemetry from nine electric vehicles used in heterogeneous driving conditions, the approach achieves faster convergence and lower prediction errors compared to single trained models. Whereas clustering solely on ambient temperature profiles produced the best results, highlighting temperature as the clearest indicator for personalized state-of-health estimation. Adding more features led to overlapping clusters and reduced model performance. Despite room for improvement in the clustering algorithm and the limited dataset, the results demonstrate promising potential. Recommended future steps include implementing richer sequence models, automated outlier filtering, and validation in larger electric vehicles fleets to provide robust, privacy-preserving battery diagnostics in real-world scenarios.

Keywords: Federated Learning, Clustering, Electric Vechiles, Battery Degradation Estimation.

Acknowledgements

This thesis and the workflow of the project could not have been carried out without the support of my supervisor at Northerly, Christian Fleisher. His guidance and continuous encouragement have been essential throughout this process.
I would also like to thank my academic supervisor, Xiaolei Bian, for his support and valuable insights from a research perspective.
Thanks as well to my examiner, Changfu Zou, for ensuring the academic quality of the work and for his helpful comments throughout the thesis process.
Last but not least, I am especially grateful to Niclas Svensson for his practical assistance with data preprocessing and help setting up the local environment—his support significantly eased the technical aspects of the project. Axel Lundkvist,

Gothenburg, June 2025

List of Acronyms

Below is the list of acronyms that have been used throughout this thesis, alphabetically listed

BMS	Battery Management System
CFL	Clustered Federated Learning
DoD	Depth of Discharge
EV	Electric Vehicle
FL	Federated Learning
ML	Machine Learning
SoH	State of Health
SoC	State of Charge
RMSE	Round Mean Square Error

Nomenclature

Below is the nomenclature of indices, sets, parameters, and variables used throughout this thesis. It reflects the terminology related to EV battery modeling, clustering, and federated learning.

Indices

i	Index for EV clients in the federated learning setup
t	Index for communication round or training iteration
s	Index for a charging session within a client dataset

Sets

$\mathcal{C}$	Set of EV clients participating in the federated learning
$\mathcal{S}_i$	Set of charging sessions for client i
$\mathcal{F}$	Set of features used in clustering (e.g., temperature, SoC, current)

Parameters

Δt	Sampling interval in telemetry data
ϵ	Silhouette threshold for triggering cluster reassignment
K	Number of clusters in K-means
K_{init}	Initial number of clusters
$K_{\max}$	Maximum number of clusters (e.g., $\lfloor N/2 \rfloor$)
s_i	Silhouette score for client i
$\bar{s}$	Mean silhouette score across all clients

Variables

$\mathbf{x}_i$	Feature vector representing usage profile of client i
X	Feature matrix for all clients, $X \in \mathbb{R}^{N \times D}$
ℓ_i	Cluster label assigned to client i
ω^t	Global model at round t
ω_m^t	Aggregated model for cluster m at round t
$\Delta\boldsymbol{\theta}_i$	Weight update from client i after local training
L_i	Local validation loss of client i

Contents

List of Figures

List of Tables

1

Introduction

1.1 Background

As the adoption of Electric Vehicles (EVs) accelerates, effective battery management remains a key challenge for manufacturers and users alike [1] [2]. Batteries naturally degrade over time due to operational stress, leading to reduced driving range, increased maintenance costs, and greater environmental impact due to replacements. Ensuring battery longevity while maintaining efficiency and safety is therefore a critical objective for sustainable EV development.

A significant limitation of current Battery Management Systems (BMS) is that they are typically designed for generalized usage patterns rather than adapting to individual driving behaviors and environmental conditions. This lack of personalization leads to suboptimal performance and inefficiencies in battery utilization.

To address these challenges, data-driven methods are increasingly used to enhance BMS functionality. Federated Learning (FL) has shown promise in battery health estimation tasks, such as remaining useful life prediction [3], while preserving data privacy. However, its application to clustering EVs based on real-world usage behavior and operational conditions remains largely unexplored. This project investigates this approach to enable more adaptive BMS strategies using both laboratory and field data.

To support the development of such adaptive systems, this project uses both controlled laboratory battery data and privacy-constrained real-world EV telemetry to explore clustering-based improvements in battery health estimation.

1.2 Purpose

The purpose of this project is to identify and achieve beneficial clustering for EV battery usage and operational conditions to obtain more precise battery metrics, within the context of a federated learning setup. This approach also aims to investigate how different feature selections impact the effectiveness of clustering for personalized battery health estimation.

1.3 Limitations

The project was carried out using a version of the FEDn framework still under active development, which introduced some constraints. For instance, the server-side logic (`server_functions.py`) supported only a minimal set of Python packages, excluding standard tools like `scikit-learn` and `pandas`. This required re-implementing all clustering and aggregation logic manually using basic Python and `numpy`.
Additionally, due to technical issues in the distributed deployment, the final experiments were conducted using a locally simulated setup. While this preserved the federated learning structure, it limited the ability to test the system's behavior in a fully distributed, multi-node environment.
Finally, due to dataset constraints, no held-out test set was used during model evaluation. All loss values reported in the results—referred to as "validation loss" for consistency—are computed on the same data used for training. As such, they reflect convergence behavior rather than true generalization performance.

2
Theory

In the following section, the theory and framework used in this project is discussed to explain how the core of the project is built.

2.1 Battery Terminology and Variables

2.1.1 State of Charge

State of Charge (SoC) is a measurement of how much charge of the maximum possible amount the battery is charged with. A SoC number is an indication of how much a battery is charged.

2.1.2 State of Health

State of health is a parameter describing the health of the battery. It says how much of the nominal capacity the battery currently can be charged with.

2.2 Lithium-ion batteries and degradation

The main battery type of commercial use is the lithium-ion battery thanks to its dense and lightweight energy capacity attribute, making it a suitable choice in the autonomous industry [4]. The lithium-ion battery is currently estimated to take a larger role in the energy sector as the number of EVs is estimated to increase with the green energy conversion [5]. There are, however, some prevalent issues in working with the component as they degrade partially due to usage stress but also due to their structural design. This means that a charged battery gradually loses both its charge and its overall capacity [6].

Some of the operations that increase their degradation are fast charging, deep discharging cycles, operating the battery in too high or too low temperatures. These are some of the mechanisms that the battery management system is trying regulate.

2.3 Battery Management Systems

A Battery Management System (BMS) is an electronic system that keeps track of the battery's health status, for instance the SoH and SoC. It controls the voltages in each cell to keep an even distributed charge while also ensuring they operate within an efficient temperature range of 20 to 40 degrees Celsius [7]. By monitoring the health, the BMS also prevents the battery from discharging too deep or overcharging which also ensures safe use of the component. However, the software has its limitations as it does not incorporate the complex dynamics of battery degradation [8].

2.4 Key Battery Degradation Features

A key consideration when clustering electric vehicles (EVs) for improved monitoring of battery State of Health (SoH) is identifying the variables that significantly influence degradation. Among these, temperature is widely recognised as one of the most critical and well-documented factors.

Numerous studies have shown that both elevated and low temperatures accelerate degradation through distinct mechanisms. At high temperatures, typically above 35°C, lithium-ion batteries experience increased rate of side reactions, including solid electrolyte interphase (SEI) growth, electrolyte decomposition, and structural damage at the electrodes. These effects contribute to capacity fade, rising impedance, and reduced cycle life [9, 10, 11].

Conversely, operation or charging at low temperatures, especially below 0°C, can lead to lithium plating on the anode. This not only results in irreversible capacity loss but also raises the risk of internal short circuits and thermal runaway if dendritic growth occurs [11, 12].

Lithium-ion batteries generally perform optimally within a moderate temperature range of 15–35°C [11, 12]. Deviations from this range, even intermittently, can lead to measurable long-term degradation. Therefore, clustering EVs based on their exposure to different ambient temperature conditions provides a meaningful strategy to identify degradation trends. This approach supports the development of personalised battery management strategies that better reflect regional and behavioural variation in EV operation.

2.5 Collaborative Learning / Federated Learning

Federated Learning (FL) is a decentralized machine learning method designed to preserve data privacy while enabling collaborative model training. Unlike traditional centralized approaches, where all data is aggregated on a single server for training, FL allows multiple clients (e.g., devices or organizations) to train local models using their own data without sharing it directly [13, 14, 15].

Each client trains a local model on its private data and periodically sends only the model updates (e.g., weights or gradients) to a central server. The server then aggregates these updates—typically using methods such as Federated Averaging (FedAvg)—to produce a global model, which is subsequently redistributed to the clients. This iterative process enables the model to learn from distributed data sources while keeping the raw data local, thus offering enhanced privacy protection and reducing the risk of data leakage.

Moreover, federated learning is well-suited to environments with heterogeneous data distributions, where client data may vary significantly. This makes FL particularly relevant for real-world applications such as mobile health, autonomous vehicles, and distributed sensor networks, where both privacy and data diversity are key concerns [16, 17].

2.6 Clustering

This section presents the theoretical background for clustering methods and evaluation metrics relevant to the analysis in this study.
In many real-world machine learning applications, labeled data is scarce or unavailable. Clustering provides a way to uncover hidden structure in unlabeled datasets by grouping data points based on similarity, making it a powerful tool in unsupervised learning. By identifying underlying categorical structures, clustering can support downstream tasks such as classification, anomaly detection, and data summarization.
Clustering algorithms vary widely in terms of their underlying principles—ranging from partitioning and hierarchical approaches to density-based and model-based methods. Each technique comes with its own strengths and limitations, and no single method is optimal for all types of data [18, 19]. The choice of clustering algorithm should therefore be guided by the characteristics of the dataset—such as dimensionality, noise, and distribution—as well as the specific goals of the analysis [20].

2.6.1 K-means

K-means is one of the most widely used clustering algorithms due to its simplicity, scalability, and efficiency. The algorithm partitions a dataset into K non-overlapping clusters by assigning each data point to the nearest mean (centroid). This process minimizes the within-cluster sum of squares (WCSS), reducing intra-cluster variance [21].

Working Principle

The algorithm operates iteratively through the following steps:

1. **Initialization:** Randomly select K initial centroids.
2. **Assignment step:** Assign each data point to the nearest centroid based on a chosen distance metric (typically Euclidean distance).
3. **Update step:** Recompute the centroid of each cluster as the mean of all points assigned to it.
4. Repeat steps 2–3 until convergence, typically defined as no change in assignments or centroids.

The objective function minimized by K-means is:

$$\sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2$$

where C_k is the set of points in cluster k, and μ_k is the centroid of that cluster.

Time Complexity

The time complexity of K-means is:

$$\mathcal{O}(n \cdot k \cdot d \cdot t)$$

where:
- n: number of data points
- k: number of clusters
- d: number of features (dimensions)
- t: number of iterations until convergence

2.6.2 Affinity Propagation (AP)

Affinity Propagation (AP) is a clustering algorithm that identifies representative examples, known as exemplars, without requiring the number of clusters to be specified in advance. It is based on the exchange of real-valued messages between data points to determine cluster membership [22].

Working Principle

AP treats each data point as a potential exemplar and exchanges two types of messages between points:
- **Responsibility** $r(i, k)$: How well-suited point k is to serve as the exemplar for point i.
- **Availability** $a(i, k)$: How appropriate it would be for point i to choose point k as its exemplar.

The algorithm updates these messages iteratively using the following rules:

$$r(i,k) \leftarrow s(i,k) - \max_{k' \neq k} \{a(i,k') + s(i,k')\}$$

$$a(i,k) \leftarrow \min\left(0, r(k,k) + \sum_{i' \notin \{i,k\}} \max(0, r(i',k))\right)$$

where $s(i, k)$ is the similarity between points i and k, typically defined as the negative squared Euclidean distance.
After convergence, exemplars are selected, and each point is assigned to the cluster of its chosen exemplar.

Time Complexity

The time complexity of AP is:

$$\mathcal{O}(n^2 \cdot t)$$

where:
- n: number of data points
- t: number of iterations

Since it operates on a full similarity matrix and updates all point-to-point relationships, AP is more computationally expensive than K-means, especially for large datasets. However, it can discover more complex structures and automatically determine the number of clusters based on data similarity and a user-defined preference parameter.

2.6.3 Silhouette Score

The Silhouette Score is a commonly used metric to evaluate the quality of clustering results, especially in unsupervised learning where true labels are unknown [23]. It provides a measure of how well each data point fits within its assigned cluster compared to other clusters, offering insight into both intra-cluster cohesion and inter-cluster separation.

For a given data point i, let:
- $a(i)$ be the average distance between i and all other points in the same cluster (intra-cluster distance).
- $b(i)$ be the minimum average distance between i and all points in any other cluster, to which i does not belong (nearest-cluster distance).

The silhouette coefficient for the point i is defined as:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

The resulting value $s(i)$ lies in the range $[-1, 1]$, where:
- $s(i) \approx 1$ indicates that the point is well clustered, with high cohesion and good separation.
- $s(i) \approx 0$ suggests that the point lies on the boundary between two clusters.
- $s(i) \approx -1$ implies that the point may have been assigned to the wrong cluster.

The overall silhouette score for the clustering result is the mean of $s(i)$ over all data points:

$$S = \frac{1}{n} \sum_{i=1}^{n} s(i)$$

A higher average silhouette score indicates better-defined clusters. This metric is especially useful when comparing different clustering algorithms or choosing the number of clusters (e.g., in K-means).

Computational Complexity

Computing the silhouette score requires calculating pairwise distances between all points to compute both $a(i)$ and $b(i)$. As a result, the time complexity is:

$$\mathcal{O}(n^2 d)$$

where n is the number of data points and d is the dimensionality of the data. This quadratic complexity can become a bottleneck for large datasets. For large-scale problems, approximate methods or sampling strategies are often used to make silhouette evaluation feasible.

3

Methods

3.1 Dataset and Preprocessing

This study employs two different datasets: the NASA Ames Prognostics Centre of Excellence dataset and an EV fleet telemetry dataset [24]. Each underwent a preprocessing pipeline tailored to its format while ensuring consistent cycle-level (NASA) or session-level (EV) features suitable for federated clustering.

3.1.1 Preprocessing Pipeline

A common four-step workflow was applied, with dataset-specific tweaks:

1. Gap Imputation via linear interpolation to maintain continuous time-series.
2. SoH Derivation and Smoothing when explicit SoH was absent (NASA).
3. Cycle/Session Demarcation using protocol flags or threshold crossings.
4. Normalisation of all features except SoH.

3.1.2 Dataset-specific Processing

NASA Dataset

Thirty-four lithium-ion cells were aged to 30 % nominal capacity at 4, 24, and 43 °C. Cells with monotonic degradation—B0029, B0030, B0046–B0048, and B0005–B0007—were selected. The long files for B0005–B0007 were split into early- and late-life segments to raise the client count and capture distinct degradation stages.

EV Fleet Telemetry Dataset

Each NDA-protected CSV file (one per vehicle) was processed as follows:

1. **Load**: parse timestamps; retain SoC, current, cell/ambient temperatures, on-board SoH.
2. **Segment**: group rows into charging sessions when SoC was non-decreasing and gaps <5 min.
3. **Synchronise & Impute**: align low-rate signals to the densest timeline; compute mean ambient temperature per session.
4. **Filter**: discard sessions with <5% SoC increase or <2 records.
5. **Feature Extraction**: session duration, initial SoH, RMS current, temperature extrema, ambient temperature.
6. **Export**: one compressed CSV per vehicle.

3.1.3 Feature Set for Clustering

Four feature groups were studied: (i) ambient temperature only, (ii) thermal (ambient and cell temperatures), (iii) usage (session duration, SoC mean and variance, and RMS current), and (iv) the combined set of all features.

3.1.4 Dataset-specific FL Data Feeding

The amount of data streamed to each client per round reflects the nature of each dataset:

- **NASA Dataset**: 20 laboratory cycles (10 charge + 10 discharge). Because the cells follow identical cycling protocols, ambient chamber temperature is the only meaningful differentiator. Therefore, clients are clustered solely on their cycles' mean ambient temperature; usage-based features are omitted. This experiment serves as a sanity-check of the FL pipeline on homogeneous, cycle-resolved data.
- **EV Fleet Dataset**: One real-world charging session. Sessions vary widely in ambient conditions, internal temperatures, charging behaviour and SoC trajectory, enabling richer thermal, usage and combined feature spaces.

These design choices allow a controlled laboratory benchmark (NASA) and a heterogeneous field scenario (EV) to be evaluated with the same FL-clustering framework, highlighting its robustness across data granularities.

3.2 Clustering Algorithm

The primary focus of this project was to come up with a clustering algorithm capable of detecting shifts in usage patterns or environmental conditions. Hence, the Dynamic Temporal Adaptive Clustered Federated Learning method ("dy-TACFL") was adopted [25], which dynamically forms client clusters based on evolving time-series features. In the authors' EV-charging setup, dy-TACFL delivered consistently reliable forecasts, indicating that its adaptability is sufficient to meet the present study's requirements. Although the original work targets station-level forecasts rather than cell-level health estimation, the method's dynamic clustering mechanism and demonstrated effectiveness render it a suitable fit for the project's needs.

The structure of the algorithm taken from the paper [25], could in a simplified manner, be explained by algorithm 1.

Algorithm 1 dy-TACFL (simplified)

Input: Rounds T, clusters K, threshold ε
Output: Final global model ω
for $t = 1$ to T **do**
  Compute silhouettes; move any i with $s_i < \varepsilon$ to its best cluster
  Build pairwise distance; run Affinity Propagation
  for each cluster C_j **do**
    Aggregate $\{\Delta\theta_i\}_{i \in C_j}$ into ω_j
  end for
  FedAvg $\{\omega_j\}$ into new ω
  for each client i **do**
    Send $\omega_{\text{cluster}(i)}$ to i
  end for
end for
Return: ω

However, since this study's work uses raw client data , whereas the original dy-TACFL paper used the model weight vector embeddings, this study has the modified the approach to algorithm 2.

Algorithm 2 Hybrid client clustering with per-round aggregation (Dy-TACFL-inspired)

Require: feature matrix $X \in \mathbb{R}^{N \times D}$, silhouette threshold ε, previous cluster count K_{prev} (take $K_{\text{prev}} = K_{\text{init}}$ on the first round)
Ensure: labels $\ell \in \{0, \dots, K-1\}^N$, centroids C

1. Pick K
if first round or K_{prev} undefined **then**
  $K \leftarrow K_{\text{init}}$
else
  $K \leftarrow K_{\text{prev}}$ ▷ reuse previous cluster count
end if
$K \leftarrow \min\Big(K, \max\{2, \lfloor N/2 \rfloor\}\Big)$ ▷ clamp to $[2, \lfloor N/2 \rfloor]$
if $K > N$ **then**
  $K \leftarrow \max\{2, \lfloor N/2 \rfloor\}$
end if
2. Baseline k-Means
$(\ell_{\text{km}}, C_{\text{km}}) \leftarrow \text{kMeans}(X, K)$
$s_i \leftarrow \text{silhouette}(X, \ell_{\text{km}})$
$U \leftarrow \{\, i \mid s_i < \varepsilon \}$
3. Affinity-Propagation refinements
if $|U| > 0.5N \;\vee\; \bar{s} < \varepsilon$ **then**
  $(\ell, C) \leftarrow \text{Affinity Propagation}(X)$
else if $|U| = 1$ **then**
  Let j be the lone index in U
  $\ell \leftarrow \ell_{\text{km}}$ with $\ell_j \leftarrow K$
  $C \leftarrow \Big[C_{\text{km}};\, X_j\Big]$
else if $|U| > 1$ **then**
  $(\hat{\ell}, \hat{C}) \leftarrow \text{AffinityPropagation}\Big(X[U]\Big)$
  Offset $\hat{\ell}$ by K and write back into ℓ_{km}
  $(\ell, C) \leftarrow \Big(\ell_{\text{km}},\, [C_{\text{km}};\, \hat{C}]\Big)$
else
  $(\ell, C) \leftarrow (\ell_{\text{km}}, C_{\text{km}})$
end if
4. Cluster-wise aggregation (FedAvg per cluster)
for each cluster C_k **do**
  $\omega_k \leftarrow \text{FedAvg}(\{\Delta\theta_i\}_{i \in C_k})$
end for
$\omega \leftarrow \text{Aggregate}(\{\omega_k\})$ ▷ Optional global merge
return $\omega_{\text{cluster}(i)}$ to each client i

After each clustering round, consistent cluster labels are maintained by matching new centroids to the nearest previous centroids in Euclidean distance using a greedy algorithm. Matched new centroids reuse their old labels, while unmatched clusters are assigned fresh IDs and obsolete clusters are removed. This stability mechanism is critical for reliably tracking client drift and interpreting temporal trends in cluster-

based metrics.
The threshold $\varepsilon = 0.50$ is adopted from the original Dy-TACFL study, where it was found to work well in their experimental setup [25]. Since the main focus of this project was on other parts of the algorithm, parametric tuning of ε was not performed, and the value from the original study was used as a reasonable default.

Symbol	**Meaning**
T	Total number of federated-learning rounds
N	Number of clients in the current round
b_i	Battery feature vector extracted by client i
D_i	Number of training examples at client i
$X \in \mathbb{R}^{N\times d}$	Feature matrix (one row per client)
K_{init}	Initial number of clusters (round 1)
$K_{\max}$	Maximum K considered (e.g. $\lfloor N/2 \rfloor$)
K	Chosen number of clusters for k-means
ℓ_{km}	Label vector from k-means ($\{1,\dots,K\}^N$)
C_{km}	Centroids from k-means ($\mathbb{R}^{K\times d}$)
s_i	Silhouette score of client i
ε	Silhouette threshold for “unstable” clients
U	Unstable client set: $\{\, i : s_i < \varepsilon\}$
$\bar{s}$	Mean silhouette: $\frac{1}{N}\sum_i s_i$
ℓ	Final label vector after AP refinement
C	Final centroids $\mathbb{R}^{K'\times d}$
ω_m^t	FedAvg model for cluster m at round t
ω^t	Global model at round t

Table 3.1: Legend of symbols used in Algorithm 2.

3.2.1 Clustering Algorithm Modifications

While the original dy-TACFL algorithm uses Affinity Propagation (AP) for clustering, running a full $O(N^2D)$ algorithm at every communication round quickly becomes computationally expensive as the number of clients N or feature dimension D grows. Moreover, using raw, high dimensional client statistics (rather than compact model updates) further increases the cost of distance computations. To address this, a hybrid approach combining standard k-Means and Affinity Propagation was developed. This preserves dy-TACFL's dynamic clustering behavior while significantly improving scalability.

Per round complexity

k-Means on N points into K clusters over I iterations costs

$$O(N\,K\,I\,D).$$

Computing silhouette scores on all N points incurs $O(N^2\,D)$ overhead each round. Affinity Propagation on the full set is $O(N^2\,D)$ but is only triggered when more than half the points fall below the silhouette threshold ε or the mean silhouette drops below ε. When applied to an unstable subset $U \subseteq \{1,\ldots,N\}$, AP costs $O(|U|^2\,D)$. Stable ID matching of the new K centroids to previous centroids costs $O(K^2\,D)$ and remains bounded by $O(N^2\,D)$ only in the worst case. However this is negligible in practice when $K \ll N$.
Taken together, the common case per round clustering cost is

$$O(N\,K\,I\,D + N^2\,D).$$

Threshold based fall backs

Silhouette scores computed on the k-Means result (Alg. 2, lines 5–7) trigger AP only when cluster quality degrades, otherwise the k-Means labels are retained. Singleton drifters and small unstable subsets invoke AP only on those points, bounding the worst case quadratic cost to $|U| < N$.

Dynamic K

An upper bound $K_{\text{max}} = \max(2, \lfloor N/2 \rfloor)$ is enforced each round. After the first round (where $K = K_{\text{init}}$), the previous round's K is reused and then clamped into $[2, \lfloor N/2 \rfloor]$, preventing both over clustering and under clustering as N varies.
Together, these modifications preserve dy-TACFL's dynamic clustering capability and forecasting accuracy, while substantially reducing average computational complexity—a critical consideration for deployment across large time varying fleets of EV chargers or battery cell health estimation sites.

3.2.2 Aggregation of cluster models

Clients were clustered based on their local data statistics, including features such as temperature and state-of-charge. Following the cluster assignment, model aggregation was performed within each cluster by computing the weighted average of the model parameters from the assigned clients. The resulting cluster models represented generalized versions of their member models and were subsequently used to update the corresponding clients in the next training round.

3.2.3 Validation of the Clustering Procedure

Step 1: Initial Sanity Check with Ambient Temperature Only

To verify basic functionality of the clustering pipeline, the first validation step tested a minimal configuration using only one feature: the mean ambient air temperature (`ambair_mean`). Despite its simplicity, this single-feature setup provides a useful baseline for assessing whether clients exposed to similar thermal environments are grouped together. Clustering was performed in each round utilizing this scalar summary per client, and results were evaluated both qualitatively (via temperature ranges within clusters) and quantitatively with the silhouette score.

Step 2: Multi-Feature Evaluation Using Silhouette Scoring

To more rigorously assess the clustering algorithm's performance, three additional multi-feature configurations were tested. These configurations combine thermal and behavioural features derived from per-client statistics over charging sessions. For each combination, clustering was performed round-wise and evaluated using the silhouette score. A threshold of 0.50 was used to indicate acceptable cluster separation. The four feature configurations used in the validation are:

- **Ambient temperature only (single feature):**
  - `ambair_mean` — mean ambient air temperature.

  This minimalist configuration tests whether a single well-behaved environmental feature can induce coherent groupings.
- **Thermal features:**
  - `ambair_mean` — mean ambient air temperature
  - `tempcell_mean` — mean of cell temperature, computed as mean$[(T_{\min} + T_{\max})/2]$
  - `tempcell_var` — variance of the same averaged cell temperature

  This group evaluates whether richer thermal descriptors improve clustering robustness and precision.
- **Usage features:**
  - `duration_min` — duration of charging session in minutes
  - `curr_rms` — root mean square of charging current
  - `soc_mean` — mean State-of-Charge during charging
  - `soc_var` — variance of State-of-Charge

  These features aim to cluster clients based on behavioural patterns, such as frequency and depth of charging.
- **Combined features:**

  `[duration_min, curr_rms, soc_mean, soc_var, ambair_mean, tempcell_mean, tempcell_var]`

  This full feature vector merges environmental and behavioural signals to evaluate whether broader context enhances clustering quality.

This four-part validation setup enables a comprehensive comparison of how different feature selections influence clustering structure, silhouette cohesion, and downstream model performance across training rounds. These were also the only available features with numeric values in the dataset—other potential categories, such as driving behavior or trip-related variables, were either missing or unavailable for analysis.

Step 3: Validation Loss as a Supplementary Evaluation Metric

In addition to the silhouette score, the mean validation loss across training rounds was monitored for each feature configuration. This metric serves to evaluate whether more meaningful or consistent clustering—indicated by higher silhouette scores—also corresponds to improved model performance. The influence of feature selection on convergence speed, final loss values, and training stability was thereby examined.

For each feature set (thermal, usage, and combined), the training process was executed multiple times to capture variability arising from stochastic elements such as model initialization and client sampling order. Validation loss was averaged across clients in each round, and the best–worst range was recorded to illustrate performance fluctuations.

This supplementary evaluation highlights the practical implications of clustering choices: feature sets that result in more coherent client groupings frequently—but not universally—leads to faster convergence and reduced variability in validation loss. Representative visualisations and analysis are presented in the results section.

Each feature was scaled using a global standardisation transform computed on all clients to ensure comparability across dimensions. Representative cluster results for each configuration are presented in the results section.

To isolate the effect of feature clustering from model complexity, a simple feedforward neural network was used throughout the evaluation. The network architecture consisted of an input layer with seven features (the seven feature that was used), followed by two hidden layers with 32 and 8 neurons respectively, each activated by ReLU functions, and a final linear output layer. This fixed model structure was applied uniformly across all experimental conditions to ensure comparability of validation loss trends.

3.3 Federated Learning Workflow and FEDn Structure

This project was initially designed to utilize the FEDn framework—developed by Scaleout Systems—to implement federated learning (FL) for battery health estimation across simulated electric vehicles. FEDn provides a distributed architecture with clients, combiners, and a controller for orchestrating model updates while preserving data privacy [26].
However, due to technical constraints and unresolved bugs in the hosted deployment, the final implementation was carried out in a locally simulated FL setup. This setup preserved the conceptual structure of the FEDn architecture but was run entirely on a single machine using standard Python scripts.
The local implementation included:

- **Orchestration (`test_script.ipynb`):** Launches a defined number of simulated EV clients and coordinates multiple FL rounds.
- **Client Logic (`simulator.py`):** Loads each client's private dataset, initializes a local model, and computes model updates.
- **Server Logic (`server_functions.py`):** Implements custom clustering and aggregation logic, distributing cluster-specific global models.

This simulation allowed iterative testing of different clustering strategies and feature combinations in a federated setting, despite not using the full FEDn deployment.

4

Results

Sections 4.1–4.4 analyse the four feature sets introduced in Section 3.2.3 Section 4.5 benchmarks a fixed-K alternative, and Sections 4.6–4.7 compare against local-only baselines as well as a laboratory sanity-check on the NASA dataset.

4.1 Clustering with Ambient Temperature Only

The following two tables showcase the algorithm's results when clustering solely on ambient temperature. Table 4.1 shows a round where the algorithm produced tighter clusters with separated ambient temperatures. In contrast, Table 4.2 shows a round with overlapping temperature ranges. For full rounds results see Appendix 1 A.

Round	Cluster	Clients	Size	Temp Range
14	0	0 (29.7), 1 (34.0), 2 (36.5), 4 (43.0), 7 (33.8), 8 (46.9)	6	[29.7–46.9]
	1	3 (8.4)	1	[8.4]
	2	5 (26.1), 6 (17.9)	2	[17.9–26.1]

Table 4.1: Example of well-separated clusters in Round 14

Round	Cluster	Clients	Size	Temp Range
18	0	1 (35.0), 4 (46.6), 5 (27.9), 8 (36.1)	4	[27.9–46.6]
	1	3 (8.9)	1	[8.9]
	2	0 (28.6), 2 (38.2), 6 (22.8), 7 (31.1)	4	[22.8–38.2]

Table 4.2: Example of overlapping clusters in Round 18

4.1.1 Evolution of Ambient Temperature-Only Clusters Over Rounds

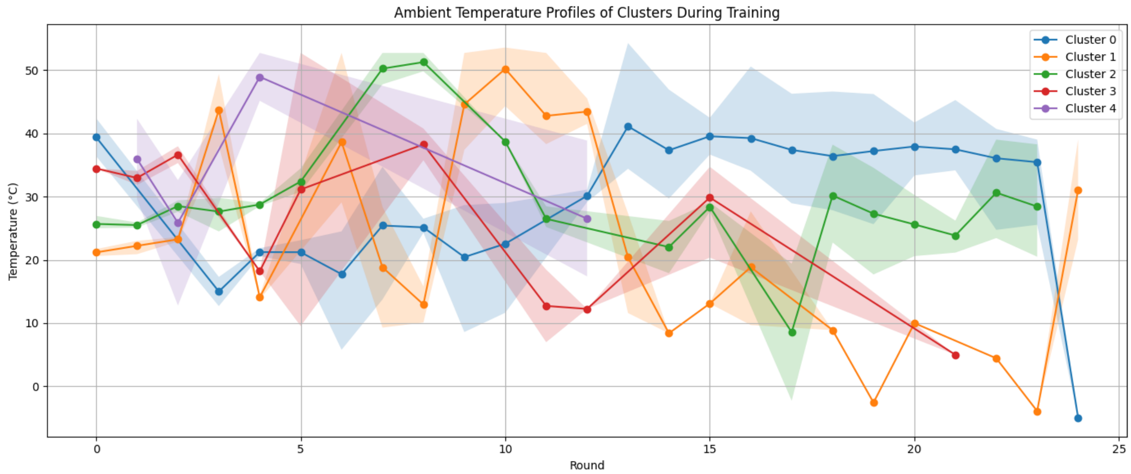

Figure 4.1: Ambient temperature profiles for each cluster across training rounds.

Figure 4.1 illustrates the ambient temperature profiles of each cluster over all training rounds. While some rounds show clear inter-cluster separation, others exhibit significant overlap in temperature ranges, confirming that ambient temperature alone can yield unstable clustering behaviour across time.

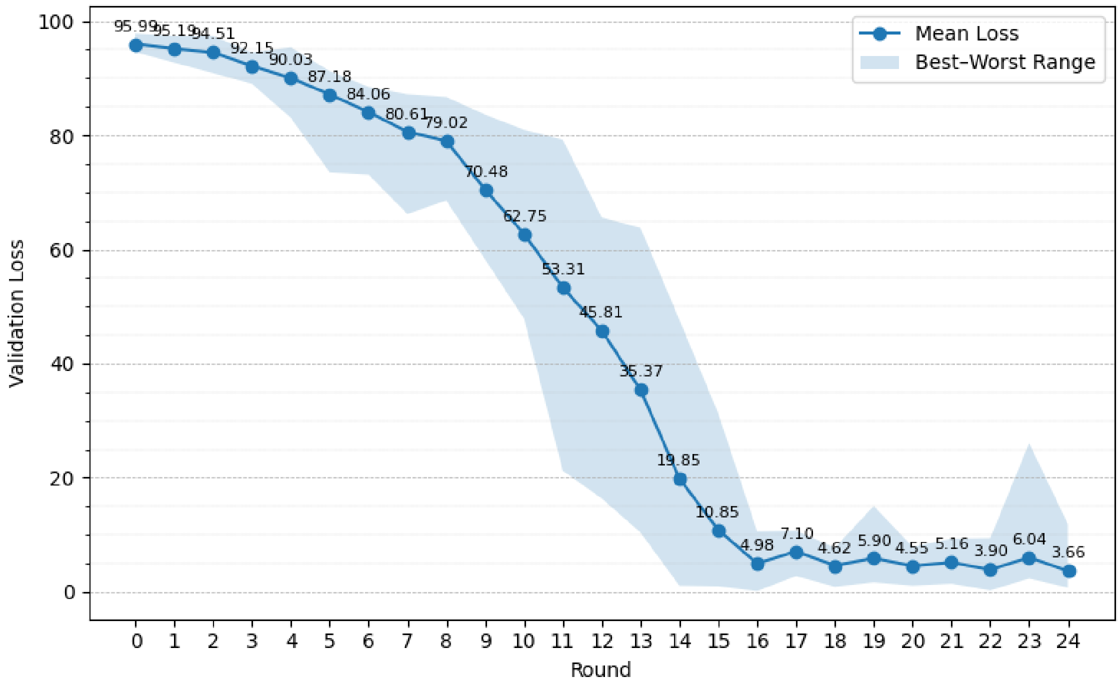

Figure 4.2: Validation result from clustering with solely ambient temperature

Figure 4.2 shows the mean validation loss across all nine clients, along with the range between the best- and worst-performing clients in each round when clustering solely based on ambient temperature. In the initial rounds (0–4), the mean validation loss remains high, hovering around 90–95, with minimal spread. From rounds 5 to 9, the loss steadily decreases from approximately 87 to 70, while the variability across clients begins to widen. A steeper drop occurs between rounds 10 and 15, with the mean loss falling sharply from about 63 to 11, accompanied by a narrowing spread, indicating convergence toward more consistent, temperature-informed clusters. After round 15, the curve flattens significantly, stabilizing between 4 and 6 by rounds 16–24, with minimal variation—demonstrating that after about 12–14 refinement rounds, the model achieves both low and robust performance, and subsequent improvements become marginal.

Round	K-means K	K-means silhouette	Final silhouette
0	3	0.7011	0.4703
1	4	0.5171	0.3924
2	4	0.5014	0.2951
3	4	0.5566	0.6272
4	4	0.7263	0.6593
5	4	0.6093	0.4780
6	4	0.3253	0.4380
7	2	0.4972	0.1416
8	3	0.5353	0.6925
9	4	0.4576	0.6088
10	2	0.6248	0.3394
11	3	0.5506	0.5398
12	4	0.5347	0.2487
13	4	0.4355	0.5710
14	2	0.5750	0.3319
15	3	0.5476	0.2406
16	4	0.3625	0.5327
17	2	0.5671	0.5671
18	3	0.4623	-0.0153
19	3	0.4704	0.0931
20	3	0.4319	0.4319
21	3	0.6115	0.6115
22	4	0.4047	0.0807
23	3	0.4417	0.1495
24	3	0.4502	0.7213

Table 4.3: Silhouette scores per round: initial K-means evaluation versus after final clustering.

The silhouette scores per round, summarized in Table 4.3, track the development of both the raw K-means clustering quality and the post-processed cluster cohesion when clustering is based solely on ambient temperature. Several trends are apparent:

- **Rounds 0–4: high initial quality, lowered by reassignment.** Raw silhouettes fall between 0.501 and 0.726, the peak occurring in Round 4. Subsequent re-assignment reduces several scores, most markedly in Round 0 (0.7011 → 0.4703), showing that clusters based on a single feature need early adjustment.

- **Rounds 5–13: greatest variability.** Final silhouettes span 0.1416 (Round 7) to 0.6925 (Round 8). In some iterations cohesion improves (Round 8: +0.1572), while in others it declines (Round 10: –0.2854), underscoring sensitivity to ambient-temperature boundaries.
- **Rounds 14–24: general convergence with two anomalies.** Most final silhouettes lie between 0.24 and 0.61, indicating gradual stabilisation. Round 18 becomes negative (–0.0153); Round 24 reaches the overall maximum (0.7213).
- **Cluster counts vary; no single value is uniformly optimal.** Across all 25 rounds, $K = 4$ occurs 11 times, $K = 3$ 10 times, and $K = 2$ 4 times. Final silhouettes above 0.60 arise with each K, so performance is not tied to a single cluster count.

In summary, clustering based solely on ambient temperature shows promising early structure, suffers some instability mid-process, but converges toward reasonably coherent and stable clusters by the later rounds.

4.2 Clustering with Thermal Features (Ambient + Cell Temperature)

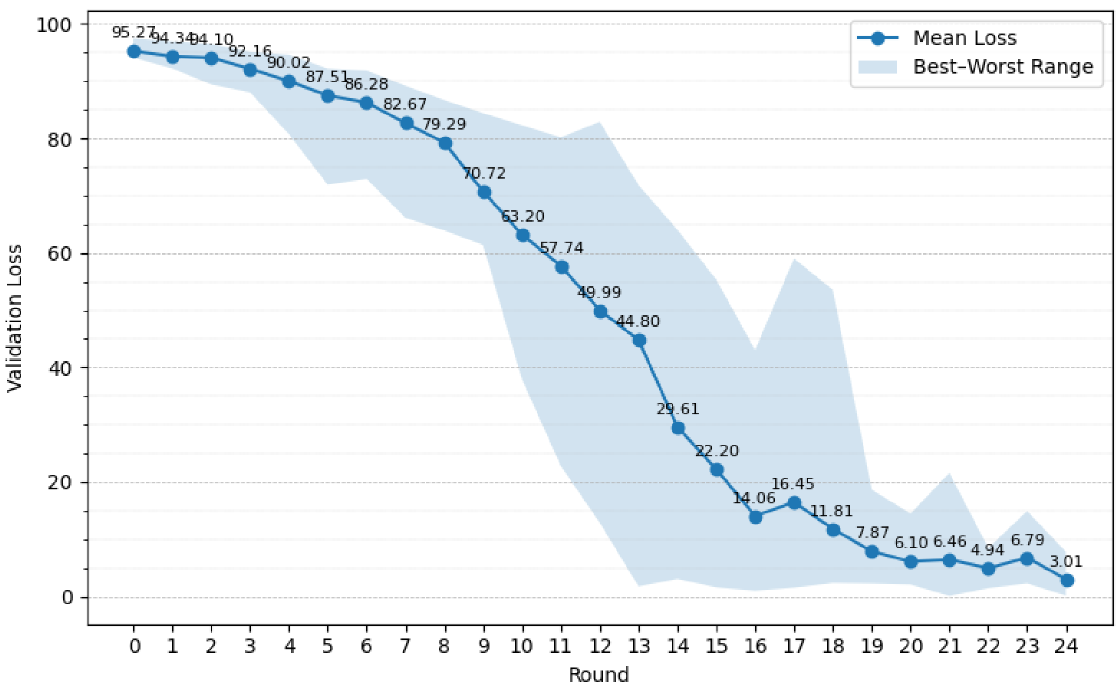

Figure 4.3: Validation result from clustering with ambient temperature and cell average temperature and its variance

Figure 4.3 plots the mean validation loss and its best–worst envelope over nine clients when clustering on both ambient air and cell temperature.
In the early phase (rounds 5–10), the average loss eases from roughly 87 down to 63, but the wide envelope shows performance still varies significantly between clients. As clustering improves (rounds 10–14), the loss drops more sharply from about 63 to 30, and the variation across clients narrows, indicating the model is converging toward the joint temperature patterns.
After round 16 the curve flattens, slowly declining to around 3–7 by rounds 20–24 with very little spread, delivering consistently low error.

Round	K-means K	K-means silhouette	Final silhouette
0	3	0.6420	0.6420
1	4	0.5171	0.3924
2	4	0.4139	−0.0653
3	3	0.6272	0.6272
4	4	0.7263	0.6593
5	4	0.6093	0.4780
6	4	0.3809	0.4380
7	2	0.4972	0.1416
8	3	0.5353	0.6925
9	4	0.4576	0.6088
10	2	0.6248	0.3394
11	3	0.5506	0.5398
12	4	0.5347	0.2487
13	4	0.5353	0.3283
14	4	0.5388	0.2785
15	4	0.5832	0.2659
16	4	0.4736	0.5327
17	2	0.5671	0.5671
18	3	0.4623	−0.0153
19	3	0.4704	0.0931
20	3	0.5160	0.4319
21	3	0.6115	0.6115
22	4	0.4047	0.0807
23	3	0.7493	0.7493
24	4	0.5752	0.4169

Table 4.4: Silhouette scores per round: initial K-means evaluation versus after final clustering.

The table in Table 4.4 reports, for each federated learning round, (1) the raw K-means silhouette score and (2) the silhouette score after final cluster assignment. A few key observations emerge:

- **Rounds 0–4:** Initial silhouettes lie between 0.414 and 0.726. Post-processing leaves the score unchanged in Rounds 0 and 3, lowers it slightly in Round 4, and markedly in Rounds 1 (0.392) and 2 (−0.065).

- **Rounds 5–12:** This interval shows the widest dispersion: final silhouettes range from 0.142 (Round 7) to 0.693 (Round 8). Refinement may either improve cohesion (Round 8: +0.158) or reduce it (Round 7: −0.355).
- **Rounds 13–24:** Raw silhouettes narrow to 0.474–0.749, and most final scores stay within ±0.15 of those values. Exceptions are Round 18 (−0.015) and Round 22 (0.081), indicating rare late-stage instability.
- **Cluster cardinality:** The algorithm selects $K = 4$ in 14 rounds, $K = 3$ in 8, and $K = 2$ in 3. Final silhouettes exceeding 0.60 occur with each value of K, so no single cluster count consistently yields superior cohesion.

Across the 25 rounds, refinement preserves or improves cohesion in 15 cases and turns negative only twice, while the strongest partitions appear in Round 4 (0.6593) and Round 23 (0.7493). These results indicate that ambient + cell-temperature features deliver generally stable, well-separated clusters, with every tested $K \in \{2, 3, 4\}$ achieving a final silhouette above 0.60 in at least one round.

4.3 Clustering with Usage Features (Duration, Current, SOC)

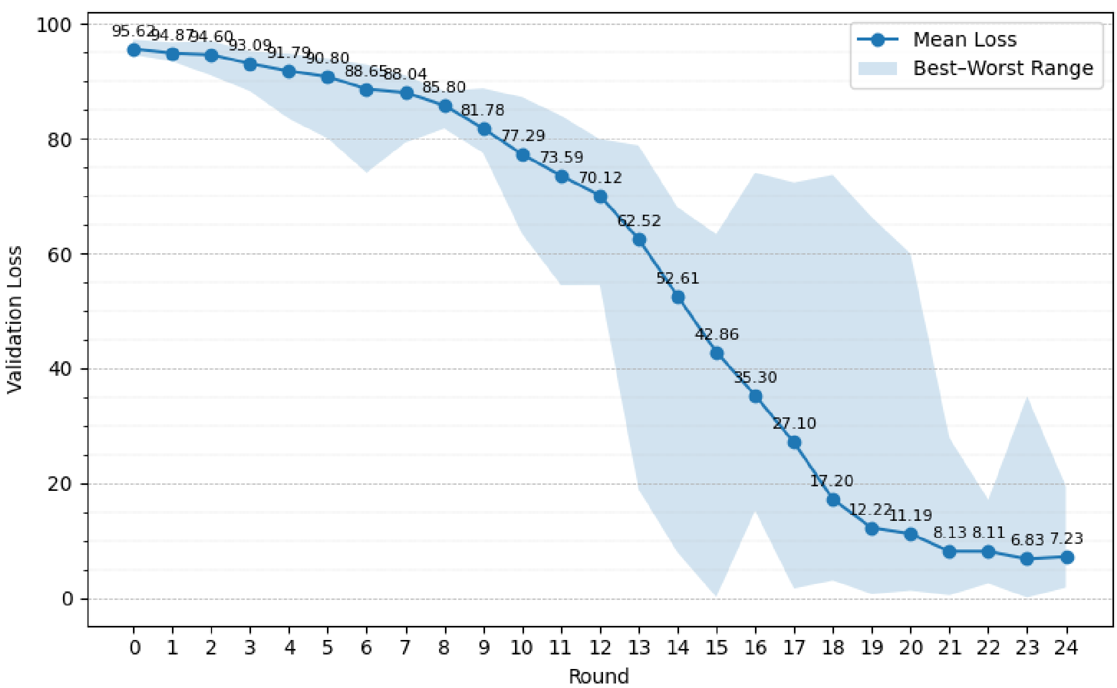

Figure 4.4: Validation result from clustering with duration of charging session, rms current of charging session and SoC average and its variance

Figure 4.4 plots the mean validation loss and its best–worst envelope over nine clients when clustering purely on charging behavior—session duration, RMS current draw, and mean state of charge.

In rounds 5–10 the mean loss declines only modestly from about 91 to 74, while the broad shaded band indicates substantial variability between clients. Between rounds 11 and 15 the loss drops steeply from roughly 74 to 43 and the envelope narrows, showing that clusters are beginning to capture the combined structure of charging duration, current and SoC.
After round 19 the curve flattens, reaching around 7–8 by rounds 20–24 with minimal spread, signalling convergence. Overall, even with these charging-specific features it takes about 19 rounds to achieve their lower errors margins. However, these are still higher than the results from the previous two feature sets.

Round	K-means K	K-means silhouette	Final silhouette
0	3	0.4283	0.3768
1	2	0.3110	-0.0877
2	3	0.2945	0.4955
3	2	0.5373	0.1769
4	3	0.3323	0.4260
5	2	0.3661	0.4301
6	3	0.3908	0.0000
7	1	0.0000	-0.0470
8	3	0.3199	-0.1125
9	3	0.4926	0.0860
10	3	0.2540	0.3567
11	2	0.2497	0.3702
12	2	0.4175	0.3963
13	2	0.4013	0.4517
14	3	0.3634	0.5604
15	2	0.3561	0.0000
16	1	0.0000	0.5499
17	2	0.2929	0.4098
18	2	0.5347	0.2944
19	3	0.5037	0.5037
20	4	0.3302	-0.0210
21	3	0.4861	0.4861
22	3	0.3459	0.0019
23	3	0.4208	-0.0782
24	3	0.4064	0.0749

Table 4.5: Silhouette scores per round for the duration + current SOC features run: initial K-means evaluation versus after final clustering.

The values summarized in Table 4.5 juxtapose the K-means silhouette with the silhouette after final cluster assignment across all 25 refinement rounds. Patterns in the table show:

- **Rounds 0–5.** Initial silhouettes span 0.294–0.537, while final values range from -0.088 to 0.496. The largest gain occurs in Round 2 (+0.201), whereas Round 1 becomes negative.

- **Rounds 6–9.** Cohesion collapses: final silhouettes reach 0.000 (Round 6) and fall below zero in Rounds 7 (-0.047) and 8 (-0.113). A limited rebound to 0.086 is observed in Round 9.
- **Rounds 10–14.** Scores stabilise and improve; final silhouettes climb from 0.357 (Round 10) to the global maximum 0.560 (Round 14) at $K = 3$.
- **Rounds 15–24.** Performance oscillates: peaks occur in Round 16 (0.550, $K = 1$), Round 19 (0.504, $K = 3$) and Round 21 (0.486, $K = 3$), whereas several rounds remain near zero or negative. Across all 25 rounds the algorithm selects $K = 3$ in 12 instances, $K = 2$ in 9, $K = 1$ in 2, and $K = 4$ once; high cohesion appears with multiple K values, indicating no single cluster count is uniformly optimal.

In summary, clustering on duration and SOC undergoes considerable early and mid-run instability before achieving its best cohesion around round 14, after which additional refinements yield uneven or limited benefit.

4.4 Clustering with Combined Feature Set

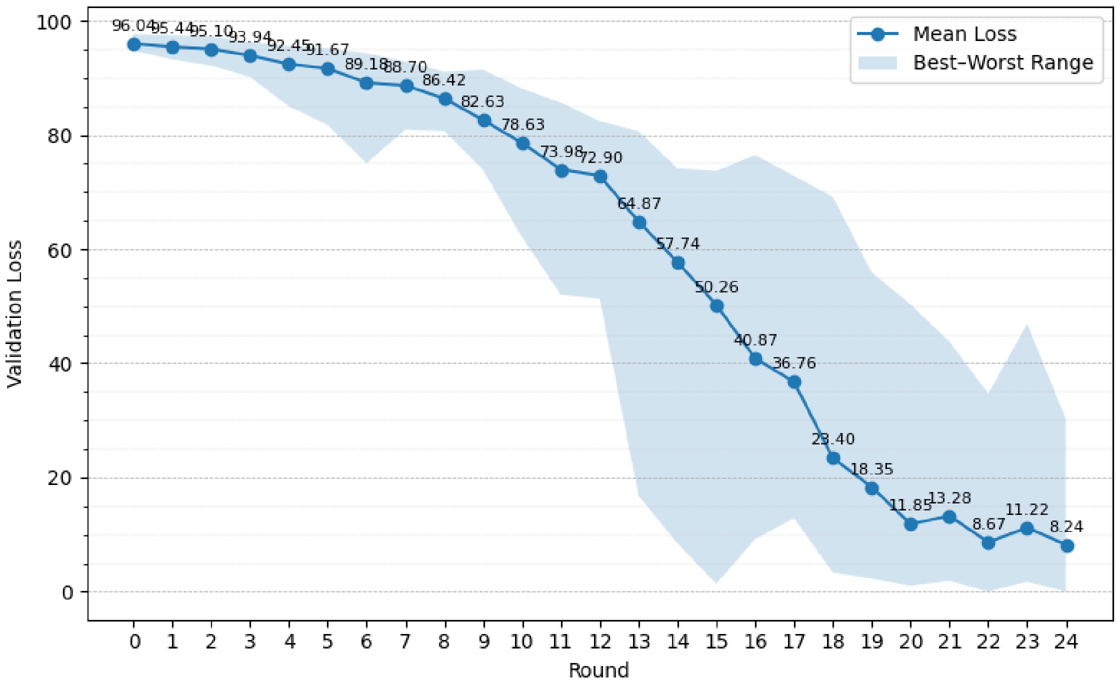

Figure 4.5: Validation loss on all features

Figure 4.5 plots the mean validation loss and its best–worst envelope over nine clients when clustering on the full feature set—ambient air temperature, cell-temperature mean and variance, charging session duration, RMS current draw, and mean state of charge.

In the early phase (rounds 5–10), the mean loss eases from about 92 down to 79, but the wide envelope makes clear that individual clients still differ markedly. The

steepest gains occur in the next stage (rounds 10–14), as loss falls sharply from roughly 79 to 58 however the shaded band grows larger, reflecting higher inconsistent, feature-informed clusters.
After round 20, the decline slows, and the loss approaches single-digit values (around 8–11 by rounds 20–23), though client variability remains high. This suggests that the model struggles to form coherent clusters when using the full feature set. Overall, these results indicate that adding more features slows convergence and reduces clustering stability and performance.

Round	K-means K	K-means silhouette	Final silhouette
0	3	0.2953	0.3497
1	2	0.4585	0.0686
2	2	0.4618	0.0660
3	2	0.4179	0.0203
4	3	0.2526	0.3378
5	3	0.4025	0.4025
6	3	0.4146	0.4237
7	2	0.5257	0.3773
8	3	0.1514	−0.0560
9	3	0.2936	0.2936
10	3	0.2073	0.3314
11	2	0.3303	0.0171
12	2	0.3154	0.2962
13	3	0.3480	−0.0268
14	4	0.1979	0.5285
15	2	0.2113	−0.0913
16	3	0.1999	0.4695
17	2	0.2193	−0.0798
18	3	0.4104	0.4104
19	3	0.3019	0.3009
20	2	0.2682	−0.0893
21	3	0.2851	−0.1098
22	3	0.2518	0.1545
23	3	0.3345	−0.0892
24	2	0.3530	0.3540

Table 4.6: Silhouette scores per round for the all-features CFL run: initial K-means evaluation versus after final clustering.

The values reported in Table 4.6 compare the raw K-means silhouette scores with their counterparts after the final clustering step for each of the 25 refinement rounds. From these results, several patterns become apparent.

- **Refinement impact.** The final clustering step raises the silhouette in 7 of 25 rounds (e.g., Round 14: +0.331), leaves it unchanged in 3, and lowers it in the remaining 15; improvement is therefore the exception rather than the norm.

- **Largest shifts.** The greatest gain occurs in Round 14 (0.198 → 0.529), followed by Round 16 (0.200 → 0.470). The sharpest declines appear in Rounds 1–3 (drops of 0.39–0.40) and Round 23 (0.335 → –0.089).
- **Overall cohesion.** Final silhouettes exceed 0.40 in only five rounds and fall below zero in seven; most values lie between 0.02 and 0.38, indicating moderate and unstable cluster quality when all features are used.
- **Cluster counts.** The procedure selects $K = 3$ in 14 rounds, $K = 2$ in 10, and $K = 4$ once. The highest silhouette (0.529) coincides with the single $K = 4$, but both strong and weak results occur for every K, suggesting that cohesion depends more on data structure than on the chosen cluster count.

Taken together, the rounds show that refinement often weakens rather than improves cohesion when all features are used, underscoring the difficulty of obtaining stable clusters.

4.5 Implemented Algorithm compared to fixed K-means

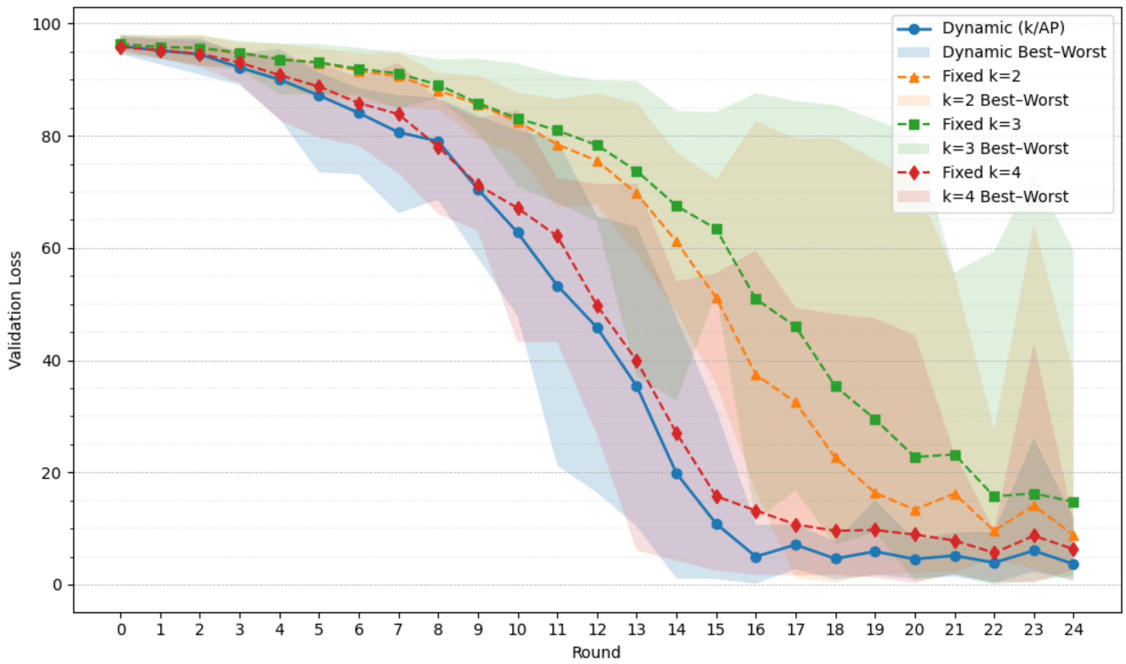

Figure 4.6: Validation loss comparison: dynamic clustering vs fixed K-means with $K = 2$, $K = 3$, and $K = 4$.

Figure 4.6 compares validation loss across training rounds for the dynamic clustering algorithm (using Affinity Propagation to determine the number of clusters adaptively) versus fixed K-means with $K = 2$, 3, and 4, using ambient temperature as the sole feature. The dynamic approach achieves lower loss faster and stabilizes around 4–6 by rounds 16–24. In contrast, all fixed-K variants converge more slowly and plateau at higher loss levels.

Furthermore, the dynamic method exhibits a narrower best–worst range from round 16 onward, indicating not only lower average loss but also more consistent performance across clients. These results highlight the benefit of allowing the number of clusters to evolve during training, especially when the optimal grouping structure is not known in advance.

4.6 Baseline: Local Training Without Federated Learning

Below are the results from two runs where clients were each given one separate data file and trained only on their local data, without any federated learning (FL) coordination.

4.6.1 Worse-Case Local-Only Training

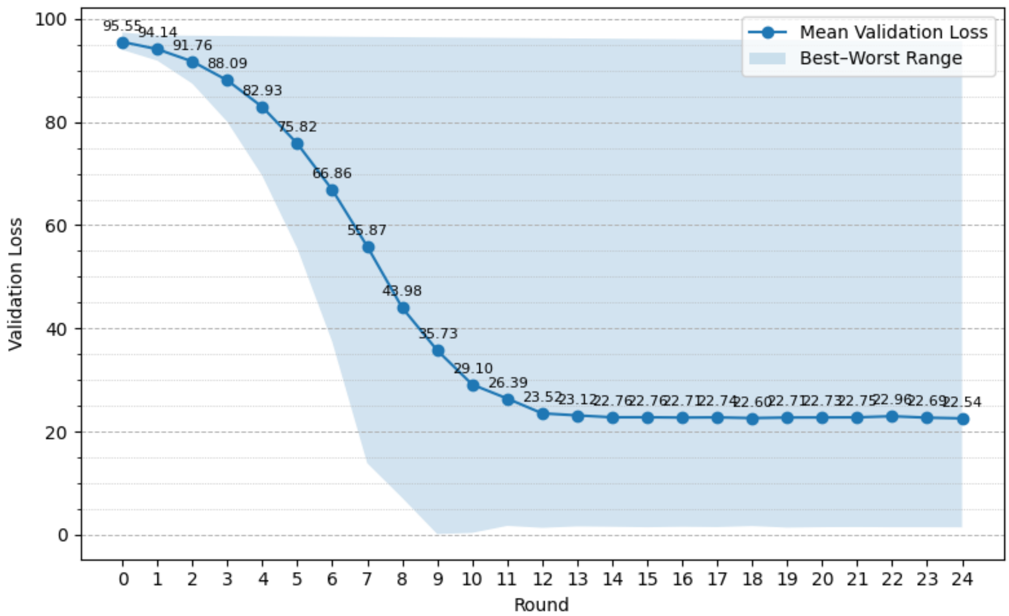

Figure 4.7: Validation loss over training rounds for the worst-case local-only training scenario (no FL). The shaded area shows the range between the best and worst performing clients, while the line indicates the mean validation loss across clients.

4.6.2 Better-Case Local-Only Training

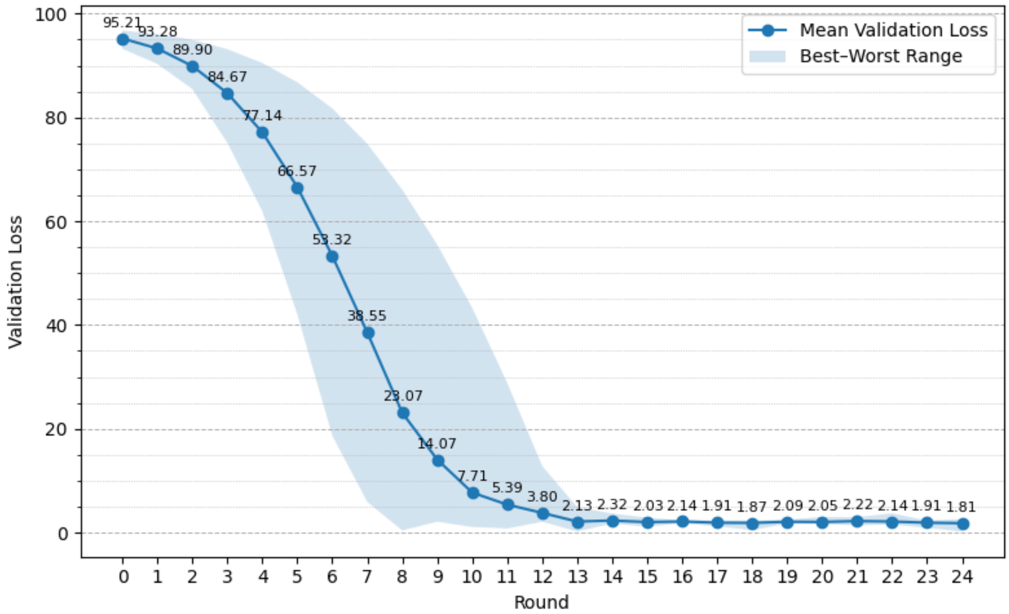

Figure 4.8: Validation loss over training rounds for the best-case local-only training scenario (no FL). The shaded area represents the best–worst range of validation loss across clients, and the line shows the mean validation loss.

Even though no changes were made to the experimental setup—the same model parameters and client datasets were used—the outcomes of local-only training vary significantly, as illustrated in Figures 4.7 and 4.8. Figure 4.7 shows a worst-case scenario where the mean validation loss converges relatively quickly but with a wide gap between the best and worst performing clients, indicating there is at least one client not learning well.
In contrast, Figure 4.8 illustrates a best-case scenario, with rapid convergence to a low validation loss (1.8) and a much narrower spread across clients after round 13. This contrast underscores the impact of non-iid data distributions across clients. While some clients benefit from representative and diverse local datasets, others underperform due to limited or skewed data. These results highlight the potential importance of collaborative training in federated learning, where shared knowledge can help mitigate such disparities and improve overall model robustness.

Supplementary Validation: NASA Dataset

Following the design in §3.3.4, the NASA benchmark clusters clients only by the mean ambient chamber temperature of their 20-cycle slice. Figure 4.9 compares the resulting FL validation loss with a local-only baseline.

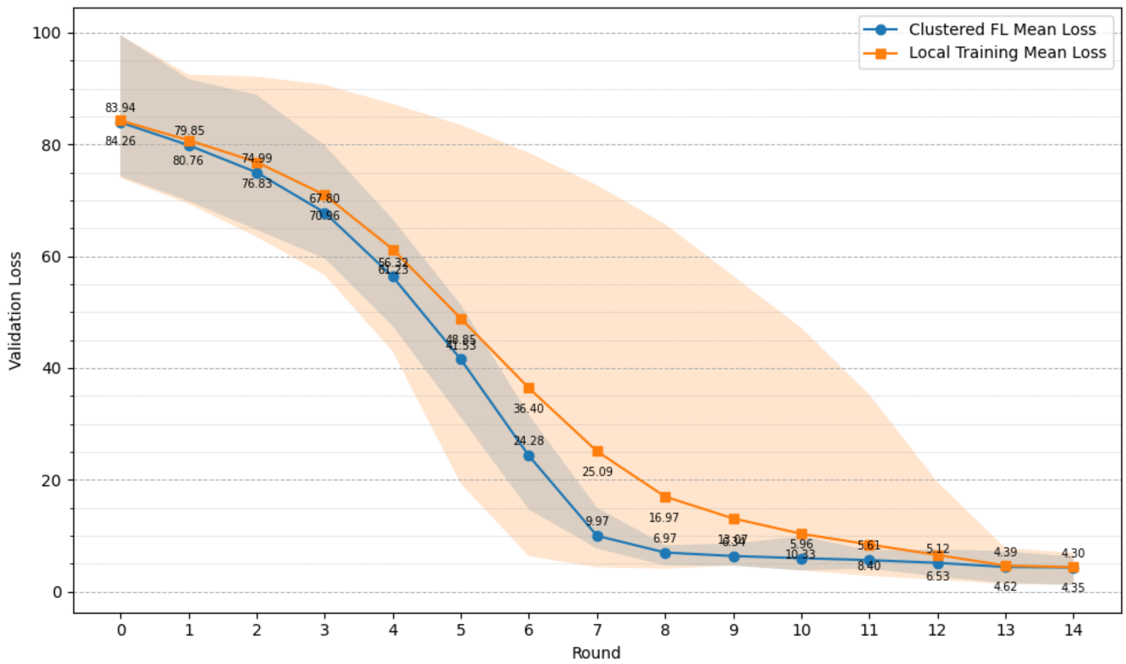

Figure 4.9: Comparison of federated learning validation loss (Clustered FL Mean Loss) versus local-only training (Local Training Mean Loss) using the NASA battery dataset across 15 training rounds. The shaded areas represent variability (best–worst range) among clients.

As illustrated in Figure 4.9, the federated learning strategy significantly reduces the validation loss compared to isolated local training. The FL model converges faster, achieving stable and consistently lower validation loss after approximately ten rounds. This comparison underscores the effectiveness of federated learning in capturing generalized battery health patterns even with cycle-level datasets, complementing the findings from the EV telemetry data.

5

Discussion

This chapter incorporate the empirical findings of Chapter 4 with the methodological choices detailed in Chapter 3, highlights the study's principal limitations, and outlines avenues for future work.

5.1 Methodological Trade-offs

5.1.1 K-means Warm-start for Affinity Propagation

Pure Affinity Propagation (AP) offers attractive properties—automatic cluster count and exemplar selection—but its $O(N^2)$ message-passing complexity makes it impractical for clustering large numbers of electric vehicles (EVs) in industrial battery-health platforms. To address this, a two-stage hybrid was adopted: (i) a K-means step ($O(NKI)$) provides initial partitions; (ii) AP selectively refines clusters based on silhouette-driven instability criteria.

While silhouette evaluation itself incurs $O(N^2D)$ cost and runs each round, full AP is only triggered when clustering quality falls below a threshold. This keeps the number of full AP invocations low and allows the hybrid pipeline to scale better than pure AP in practice. As shown in Fig. 4.6, the approach converges faster and achieves lower validation loss than the fixed-K K-means baseline, while remaining computationally manageable for mid-sized EV fleets.

5.2 Interpretation of Key Findings

5.2.1 Ambient-Temperature Dominance

Across all feature sets, the strongest and most stable clusters arose when ambient temperature alone was used as the discriminative attribute Figure 4.2. This pattern holds even more starkly in the NASA sanity-check (Fig. 4.9), where ambient chamber temperature is the only source of heterogeneity. This suggests that, in this study, CFL was most effective when the clients shared at least one clear and consistent pattern. In the EV data, the ways the cars are charged differ so much that adding more features (session length, RMS current, SoC figures) just spreads the clients further apart. As a result, the silhouette scores more often fall than rise after the AP refinement step (see Table 4.6). It should be noted, however, that the current clustering pipeline—particularly the reliance on full silhouette evaluation and the lack of feature scaling or weighting—may not fully capture the latent structure in client behaviour. These implementation choices likely affect the quality of the clustering, and thus limit the strength of any conclusions drawn about feature importance or CFL effectiveness.

5.2.2 Model Complexity and Limited CFL Gain

The Dy-TACFL reference study employed stacked LSTM encoders on multi-signal current profiles and reported marked CFL gains. By contrast, the present work uses a shallow feed-forward network that maps a single charging session to an SoH estimate. With such a low-capacity model the representational gap between clients is small, and the marginal utility of sharing gradients is likewise limited. This could partly explain why CFL outperforms local-only training (Fig. 4.9) yet shows only modest advantage over the ambient-only baseline.

5.3 Limitations

5.3.1 Sensor and Label Noise

Approximately 18% of EV sessions report cell or ambient temperatures exceeding 40 °C. While such values are plausible during aggressive fast charging or in hot climates, they may also reflect sensor drift or faulty readings. Without additional metadata (e.g., cooling status, location, or vehicle diagnostics), it is difficult to distinguish between valid extremes and measurement noise. These high-temperature entries widen intra-cluster variance (see Table 4.2) and may lead Affinity Propagation to over-split otherwise cohesive groups. Future pipelines should consider uncertainty-aware clustering or robust pre-filtering mechanisms that can adapt to context, rather than applying fixed thresholds.

5.3.2 Software and Runtime Constraints

Early versions of the FEDn deployment suffered from unstable interactions between the controller and the combiner, which disrupted training workflows and limited the pace of experimentation. Several runs had to be repeated due to runtime errors, which reduced the time available for exploring alternative clustering strategies or tuning hyperparameters. All results in Chapter 4 are drawn from the final, stable, fully local deployment.

5.3.3 Data Scarcity and Usage Heterogeneity

Each EV client contributes only one charging session per FL round. This constraint was imposed deliberately to ensure that all vehicles participated in the same number of rounds and had comparable data volume, given the limited session length of many clients in the dataset. While this uniformity improves fairness in training, it also introduces data sparsity that limits the statistical power of inter-client gradient aggregation. Combined with the pronounced diversity in driving and charging behaviours, this likely caps the SoH accuracy improvements obtainable through CFL in this setting.

5.4 Future Work

- **Adaptive AP tuning.** Automating the preference parameter and introducing multi-threshold or hierarchical AP could yield finer-grained, high-cohesion clusters without manual inspection.
- **Richer models and signals.** Integrating drive-cycle statistics, fast-charge flags and recurrent (GRU/LSTM) encoders should amplify the benefit of CFL, as suggested by prior work.
- **Robust data-quality filters.** Outlier rejection and temperature-sensor calibration would stabilise clustering and silhouette trajectories.
- **Large-scale validation.** Running the hybrid K-means plus AP pipeline on $> 1\,000$ field vehicles would test its claimed scalability and uncover communication bottlenecks.
- **Efficient silhouette approximations.** Sub-sampling or cluster-level caching could reduce the current $O(N^2D)$ cost to near-linear time, enabling more scalable clustering.

6 Conclusion

This thesis presents a dynamic federated clustering framework for estimating the State of Health (SoH) of electric vehicle (EV) batteries, combining K-means warm-starts with Affinity Propagation to enable adaptive client grouping. While the approach is computationally tractable and conceptually promising, the empirical results highlight clear limitations in its current implementation.

Most notably, the algorithm often struggled to form stable and meaningful clusters when provided with richer feature sets, such as charging behavior metrics. In many cases, silhouette scores declined after refinement, suggesting that the clustering logic is not yet fully reliable across heterogeneous usage patterns. The consistently better results observed when clustering on ambient temperature alone raise important questions—not just about the strength of that single feature, but also about the algorithm's ability to handle multidimensional data effectively.

These findings suggest that while ambient temperature appears to dominate cluster formation in this setup, this may reflect algorithmic sensitivity or suboptimal parameterization rather than true feature superiority. Therefore, a key direction for future work is to revisit and refine the clustering logic—potentially incorporating improved dimensionality handling, dynamic threshold tuning, or alternative clustering methods—to more rigorously determine whether ambient temperature should indeed remain the primary driver of personalization in battery health estimation.

In its current form, the federated clustering pipeline provides modest but consistent gains over local-only baselines, especially in controlled settings like the NASA benchmark. However, to scale effectively in real-world, diverse EV fleets, the framework must evolve. Enhancing model complexity, introducing more robust feature selection, and refining clustering dynamics will be essential for realizing the full potential of adaptive federated learning battery analytics.

Bibliography

[1] M. Berecibar, M. Garmendia, I. Gandiaga, J. Crego, and I. Villarreal, "State of health estimation algorithm of lifepo4 battery packs based on differential voltage curves for battery management system application," *Energy*, vol. 103, pp. 784–796, 5 2016.

[2] H. Rahimi-Eichi, U. Ojha, F. Baronti, and M.-Y. Chow, "Battery management system: An overview of its application in the smart grid and electric vehicles," *IEEE Industrial Electronics Magazine*, vol. 7, no. 2, pp. 4–16, 2013.

[3] R. Zhong, B. Hu, Y. Feng, S. Lou, Z. Hong, F. Wang, G. Li, and J. Tan, "Lithium-ion battery remaining useful life prediction: a federated learning-based approach," *Energy, Ecology and Environment*, 10 2024, questions:How do they conduct RUL estimation of the Li-ion batteries when the data is isolated?What dataset are they using?How can you use this article in your thesis/paper/for Christian?

[4] Clean Energy Institute, University of Washington, "Lithium-ion battery," https://www.cei.washington.edu/research/energy-storage/lithium-ion-battery/, n.d., accessed: 2025-06-04.

[5] International Energy Agency, "Global ev outlook 2023," https://www.iea.org/reports/global-ev-outlook-2023/trends-in-batteries, 2023, accessed: 2025-06-04.

[6] A. Wang, S. Kadam, H. Li, S. Shi, and Y. Qi, "Review on modeling of the anode solid electrolyte interphase (sei) for lithium-ion batteries," 12 2018.

[7] Y. Ortiz, P. Arévalo, D. Peña, and F. Jurado, "Recent advances in thermal management strategies for lithium-ion batteries: A comprehensive review," 3 2024.

[8] P. Rahmani, S. Chakraborty, I. Mele, T. Katrašnik, S. Bernhard, S. Pruefling, S. Wilkins, and O. Hegazy, "Driving the future: A comprehensive review of automotive battery management system technologies, and future trends," *Journal of Power Sources*, vol. 629, p. 235827, 2 2025. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0378775324017798

[9] S. Ma, M. Jiang, P. Tao, C. Song, J. Wu, J. Wang, T. Deng, and W. Shang, "Temperature effect and thermal impact in lithium-ion batteries: A review," pp. 653–666, 12 2018.

[10] F. Leng, C. M. Tan, and M. Pecht, "Effect of temperature on the aging rate of li ion battery operating above room temperature," *Scientific Reports*, vol. 5, 8 2015.

[11] S. Chen, X. Wei, G. Zhang, X. Wang, J. Zhu, X. Feng, H. Dai, and M. Ouyang, "All-temperature area battery application mechanism, performance, and strategies," 7 2023.

[12] J. Bao, Y. Mao, Y. Zhang, H. Xu, Y. Jiang, and Y. Yang, "Critical review of temperature prediction for lithium-ion batteries in electric vehicles," 12 2024.

[13] 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

[14] J. Konečný, H. B. McMahan, D. Ramage, and P. Richtárik, "Federated optimization: Distributed machine learning for on-device intelligence," 2016. [Online]. Available: https://arxiv.org/abs/1610.02527

[15] A. Z. Tan, H. Yu, L. Cui, and Q. Yang, "Towards personalized federated learning," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, pp. 9587–9603, 12 2023.

[16] P. Kairouz, H. B. McMahan *et al.*, "Advances and open problems in federated learning," *arXiv preprint arXiv:1912.04977*, 2019.

[17] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Processing Magazine*, vol. 37, no. 3, pp. 50–60, 2020.

[18] A. K. Jain, "Data clustering: 50 years beyond k-means," *Pattern Recognition Letters*, vol. 31, pp. 651–666, 6 2010.

[19] D. Xu and Y. Tian, "A comprehensive survey of clustering algorithms," *Annals of Data Science*, vol. 2, pp. 165–193, 6 2015.

[20] A. eddy, "Data clustering data clustering algorithms and applications chapman & hall/crc data mining and knowledge discovery series chapman & hall/crc data mining and knowledge discovery series," Tech. Rep., 2014.

[21] S. Lloyd, "Least squares quantization in pcm," *IEEE Transactions on Information Theory*, vol. 28, no. 2, pp. 129–137, 1982.

[22] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.

[23] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.

[24] B. Saha and K. Goebel, "Battery Data Set," NASA Ames Research Center, NASA Prognostics Data Repository, Moffett Field, CA, Tech. Rep., 2007, online. Available: https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/ (accessed 2025-05-06).

[25] S. S. Ali, M. Ali, D. M. S. Bhatti, and B. J. Choi, "dy-tacfl: Dynamic temporal adaptive clustered federated learning for heterogeneous clients," *Electronics (Switzerland)*, vol. 14, 1 2025.

[26] M. Ekmefjord, A. Ait-Mlouk, S. Alawadi, M. Åkesson, D. Stoyanova, O. Spjuth, S. Toor, and A. Hellander, "Scalable federated machine learning with fedn," *arXiv preprint arXiv:2103.00148*, 2021.

A

Appendix 1

Round 0

Round	Cluster	Clients	Temp Range
0	0	client: 0 (T=36.4), 1 (T=42.4)	[36.4–42.4]
0	1	client: 4 (T=21.2), 5 (T=21.7), 7 (T=20.6)	[20.6–21.7]
0	2	client: 3 (T=27.0), 6 (T=25.1), 8 (T=25.0)	[25.0–27.0]
0	3	client: 2 (T=34.5)	[34.5]

Round 1

Round	Cluster	Clients	Temp Range
1	1	client: 4 (T=22.8), 7 (T=20.9), 8 (T=23.0)	[20.9–23.0]
1	2	client: 3 (T=25.9), 6 (T=25.1)	[25.1–25.9]
1	3	client: 0 (T=34.0), 1 (T=31.9)	[31.9–34.0]
1	4	client: 2 (T=42.3), 5 (T=29.6)	[29.6–42.3]

Round 2

Round	Cluster	Clients	Temp Range
2	1	client: 3 (T=22.9), 6 (T=23.6)	[22.9–23.6]
2	2	client: 1 (T=29.4), 4 (T=27.6)	[27.6–29.4]
2	3	client: 0 (T=38.0), 2 (T=35.3)	[35.3–38.0]
2	4	client: 5 (T=32.6), 7 (T=12.8), 8 (T=32.1)	[12.8–32.6]

Round 3

Round	Cluster	Clients	Temp Range
3	0	client: 6 (T=17.3), 7 (T=12.7)	[12.7–17.3]
3	1	client: 4 (T=38.0), 8 (T=49.3)	[38.0–49.3]
3	2	client: 0 (T=27.4), 1 (T=28.4), 2 (T=28.1), 3 (T=24.5), 5 (T=29.8)	[24.5–29.8]

Round 4

Round	Cluster	Clients	Temp Range
4	0	client: 2 (T=20.6), 3 (T=21.9)	[20.6–21.9]
4	1	client: 0 (T=14.4), 7 (T=13.8)	[13.8–14.4]
4	2	client: 1 (T=28.5), 5 (T=29.0)	[28.5–29.0]
4	3	client: 6 (T=18.2)	[18.2]
4	4	client: 4 (T=52.7), 8 (T=45.1)	[45.1–52.7]

Round 5

Round	Cluster	Clients	Temp Range
5	0	client: 0 (T=22.0), 2 (T=23.1), 6 (T=19.4), 7 (T=20.4)	[19.4–23.1]
5	2	client: 1 (T=32.0), 5 (T=30.8), 8 (T=34.5)	[30.8–34.5]
5	3	client: 3 (T=9.6), 4 (T=52.7)	[9.6–52.7]

Round 6

Round	Cluster	Clients	Temp Range
6	0	client: 0 (T=19.5), 2 (T=24.5), 3 (T=5.8), 6 (T=16.0), 7 (T=22.9)	[5.8–24.5]
6	1	client: 1 (T=31.8), 4 (T=52.7), 5 (T=29.1), 8 (T=40.9)	[29.1–52.7]

Round 7

Round	Cluster	Clients	Temp Range
7	0	client: 0 (T=23.0), 1 (T=33.7), 2 (T=34.7), 6 (T=13.8), 7 (T=21.9)	[13.8–34.7]
7	1	client: 3 (T=9.3), 5 (T=28.4)	[9.3–28.4]
7	2	client: 4 (T=52.7), 8 (T=47.8)	[47.8–52.7]

Round 8

Round	Cluster	Clients	Temp Range
8	0	client: 0 (T=24.6), 5 (T=26.5), 7 (T=24.3)	[24.3–26.5]
8	1	client: 3 (T=10.1), 6 (T=15.8)	[10.1–15.8]
8	2	client: 2 (T=49.8), 4 (T=52.7)	[49.8–52.7]
8	3	client: 1 (T=35.8), 8 (T=40.8)	[35.8–40.8]

Round 9

Round	Cluster	Clients	Temp Range
9	0	client: 0 (T=28.6), 3 (T=8.6), 5 (T=26.4), 6 (T=16.5), 7 (T=22.1)	[8.6–28.6]
9	1	client: 1 (T=37.4), 2 (T=43.8), 4 (T=52.7), 8 (T=44.4)	[37.4–52.7]

Round 10

Round	Cluster	Clients	Temp Range
10	0	client: 0 (T=29.0), 3 (T=11.6), 5 (T=26.8), 6 (T=18.1), 7 (T=27.2)	[11.6–29.0]
10	1	client: 2 (T=44.3), 4 (T=52.7), 8 (T=53.5)	[44.3–53.5]
10	2	client: 1 (T=38.7)	[38.7]

Round 11

Round	Cluster	Clients	Temp Range
11	1	client: 1 (T=40.5), 2 (T=39.5), 4 (T=52.7), 8 (T=38.3)	[38.3–52.7]
11	2	client: 0 (T=25.8), 5 (T=25.2), 7 (T=28.5)	[25.2–28.5]
11	3	client: 3 (T=7.0), 6 (T=18.5)	[7.0–18.5]

Round 12

Round	Cluster	Clients	Temp Range
12	0	client: 5 (T=29.1), 7 (T=31.1)	[29.1–31.1]
12	1	client: 2 (T=41.5), 4 (T=43.3), 8 (T=45.5)	[41.5–45.5]
12	3	client: 3 (T=12.2)	[12.2]
12	4	client: 0 (T=23.3), 1 (T=38.8), 6 (T=17.4)	[17.4–38.8]

Round 13

Round	Cluster	Clients	Temp Range
13	0	client: 1 (T=34.4), 2 (T=37.0), 4 (T=38.8), 8 (T=54.2)	[34.4–54.2]
13	1	client: 0 (T=22.5), 3 (T=11.6), 5 (T=23.1), 6 (T=17.5), 7 (T=27.4)	[11.6–27.4]

Round 14

Round	Cluster	Clients	Temp Range
14	0	client: 0 (T=29.7), 1 (T=34.0), 2 (T=36.5), 4 (T=43.0), 7 (T=33.8), 8 (T=46.9)	[29.7–46.9]
14	1	client: 3 (T=8.4)	[8.4]
14	2	client: 5 (T=26.1), 6 (T=17.9)	[17.9–26.1]

Round 15

Round	Cluster	Clients	Temp Range
15	0	client: 2 (T=36.6), 4 (T=42.5), 8 (T=39.5)	[36.6–42.5]
15	1	client: 3 (T=13.1)	[13.1]
15	2	client: 5 (T=27.6), 7 (T=29.1)	[27.6–29.1]
15	3	client: 0 (T=34.7), 1 (T=34.5), 6 (T=20.4)	[20.4–34.7]

Round 16

Round	Cluster	Clients	Temp Range
16	0	client: 0 (T=34.6), 1 (T=39.2), 2 (T=34.5), 4 (T=42.6), 7 (T=34.1), 8 (T=50.5)	[34.1–50.5]
16	1	client: 3 (T=9.7), 5 (T=27.6), 6 (T=19.2)	[9.7–27.6]

Round 17

Round	Cluster	Clients	Temp Range
17	0	client: 0 (T=31.5), 1 (T=45.3), 2 (T=35.0), 4 (T=46.2), 5 (T=29.0), 7 (T=32.0), 8 (T=42.7)	[29.0–46.2]
17	2	client: 3 (T=-2.2), 6 (T=19.4)	[-2.2–19.4]

Round 18

Round	Cluster	Clients	Temp Range
18	0	client: 1 (T=35.0), 4 (T=46.6), 5 (T=27.9), 8 (T=36.1)	[27.9–46.6]
18	1	client: 3 (T=8.9)	[8.9]
18	2	client: 0 (T=28.6), 2 (T=38.2), 6 (T=22.8), 7 (T=31.1)	[22.8–38.2]

Round 19

Round	Cluster	Clients	Temp Range
19	0	client: 0 (T=36.2), 2 (T=36.5), 4 (T=46.2), 5 (T=25.6), 8 (T=41.4)	[25.6–46.2]
19	1	client: 3 (T=-2.5)	[-2.5]
19	2	client: 1 (T=34.5), 6 (T=17.7), 7 (T=29.6)	[17.7–34.5]

Round 20

Round	Cluster	Clients	Temp Range
20	0	client: 0 (T=40.7), 1 (T=34.5), 2 (T=39.3), 4 (T=41.7), 8 (T=33.4)	[33.4–41.7]
20	1	client: 3 (T=10.0)	[10.0]
20	2	client: 5 (T=26.0), 6 (T=20.6), 7 (T=30.2)	[20.6–30.2]

Round 21

Round	Cluster	Clients	Temp Range
21	0	client: 0 (T=37.5), 1 (T=35.1), 2 (T=34.1), 4 (T=45.2), 8 (T=35.4)	[34.1–45.2]
21	2	client: 5 (T=26.2), 6 (T=21.1), 7 (T=24.2)	[21.1–26.2]
21	3	client: 3 (T=5.0)	[5.0]

Round 22

Round	Cluster	Clients	Temp Range
22	0	client: 0 (T=39.2), 1 (T=40.7), 2 (T=38.4), 4 (T=37.2), 5 (T=24.8)	[24.8–40.7]
22	1	client: 3 (T=4.5)	[4.5]
22	2	client: 6 (T=23.4), 7 (T=29.4), 8 (T=39.0)	[23.4–39.0]

Round 23

Round	Cluster	Clients	Temp Range
23	0	client: 0 (T=39.0), 2 (T=37.0), 4 (T=37.2), 7 (T=25.5), 8 (T=38.5)	[25.5–39.0]
23	1	client: 3 (T=-3.9)	[-3.9]
23	2	client: 1 (T=38.2), 5 (T=26.5), 6 (T=20.5)	[20.5–38.2]

Round 24

Round	Cluster	Clients	Temp Range
24	0	client: 3 (T=-5.0)	[-5.0]
24	1	client: 0 (T=39.0), 1 (T=37.2), 2 (T=32.7), 4 (T=32.5), 5 (T=25.0), 6 (T=23.2), 7 (T=29.0), 8 (T=29.6)	[23.2–39.0]

DEPARTMENT OF ELECTRICAL ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

CHALMERS
UNIVERSITY OF TECHNOLOGY