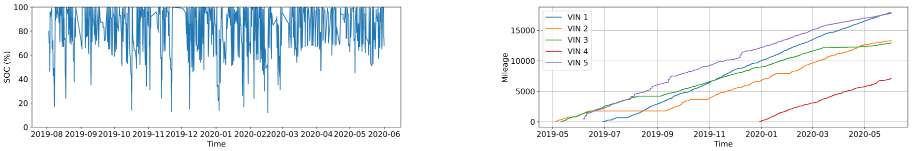
EXPLORING BATTERY USAGE IN ELECTRIC VEHICLES THROUGH GRAPH BASED CASCADED CLUSTERING

Anonymous authors
Paper under double-blind review

A APPENDIX

A.1 DATA

Plot 1a visualizes the SOC parameter over the lifetime of a single VIN, with the value oscillating between 0 and 100% as the vehicle is charged and driven. Plot 1b visualizes the mileage over time for five different vehicles, showing a sample of the range of driving frequencies present in the dataset.



(a) SOC vs Time for a Single VIN

(b) Mileage vs Time for Five Selected VINs

Figure 1: Summary of raw data characteristics and visualization

A.2 FEATURE ENGINEERING

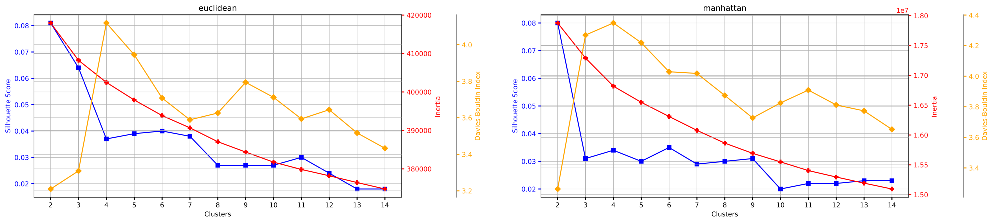
We summarize all the extra features we derived in Table 1. A discharge cycle is defined when max SOC comes before min SOC, while a charging cycle is defined when min SOC comes before max SOC. We also define a feature called “weekly cycle”, that takes into account the number of charge/discharge cycles in a specific week cumulatively with every measurements. This encodes implicitly the duration of charging and discharging cycles in a week.

Table 1: Domain-specific derived features

Feature	**Formula**	**Description**
ΔSOC	$\text{SOC}_t - \text{SOC}_{t-1}$	the change in SOC of a vehicle, taking into consideration the previous SOC
weekly mile	$\sum_0^M \text{mile}_\text{i}$	sum up the mileage through all measurements M within a week
Δmile	$\text{mile}_t - \text{mile}_{t-1}$	change in mileage among sequential measurements
depth of discharge	$\text{SOC}_\text{max} - \text{SOC}_\text{min}$	the amount the battery discharges during a drive/discharge cycle (d-cycle)
ΔEnergy	$\Delta\text{SOC} * \frac{\text{E}_\text{thermal}}{\text{E}_\text{rating}}$	the measure of how much energy has been gained or lost by the battery. $\frac{\text{E}_\text{thermal}}{\text{E}_\text{rating}}$ is a constant
Velocity	$\Delta\text{mile} * \frac{60}{10(\text{t}_\text{interval})}$	velocity is calculated as distance with respect to time. In our case, 10 min intervals

A.3 CLUSTERING STAGE-ONE EXPERIMENTS

To thoroughly assess the clustering algorithms’ performance on our data, we initially applied them to various K-values using a fixed random seed of 0. Figure 2 shows the three internal validation metrics namely, Inertia, Silhouette score and D.B. Index for K-medoids over all three distance metrics for different K-values. In the case of Agglomerative clustering, we first adopt a distance threshold estimation but it does not allow for equal comparison, since the K-values vary significantly. Hence, we apply the same set of experiments to Agglomerative clustering in order to compare all the algorithms on the same set of K-values. Table 2 contains a subset of experiments we ran using our best estimates from our strategy for choosing K using the same random seed (0).



(a) a

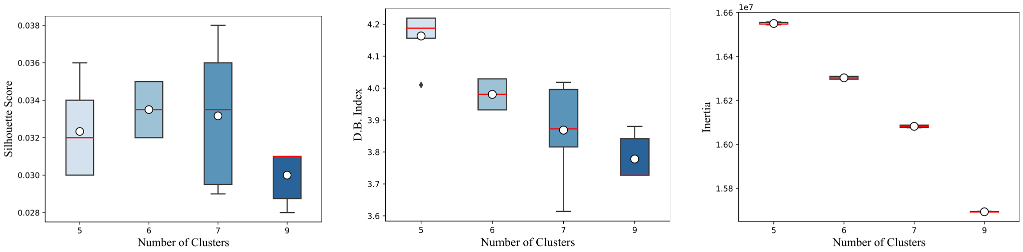
(b) b

Figure 2: Choosing K plots for Level-1 Clustering

Table 2: Scores for stage-one clustering experiments

Algorithm	No. of Clusters (K)	Distance Metric	Silhouette score	Inertia (Sum of Distances)	Davies-Bouldin index
K-medoids	6	Manhattan	0.035	16309816.114	4.029
	6	Euclidean	0.040	393856.804	3.708
K-means	5	Euclidean	0.055	298.107	2.966
	6	Euclidean	0.054	294.291	3.310
Agglomerative	6	Euclidean	0.024	-	3.968
	7	Euclidean	0.023	-	3.738
	9	Euclidean	0.022	-	3.742

Leveraging our estimation of K from Figure 2, we conduct a random seed experiment for a subset of K-values k ∈ { 5, 6, 7, 9 }. For different combinations of clustering algorithm and distance metric, we perform the experiment using random seeds from the set { 0, 1, 2, 3, 4, 5 }. This comprehensive approach allows us to explore the effectiveness of different configurations and identify the most suitable combinations for further analysis. The error bars for K-medoids algorithm with Manhattan distance are shown in Fig 3. The statistics of error in silhouette score for all three clustering algorithms is given in Table 3. Each box in Figure 3a shows the quartiles of the Silhouette scores of different random seeds at different values of K, while the whiskers extend to show the rest of its distribution. The red line represents the median value and the circle represents the mean. For K = 6, we see that the mean and median are the same. This shows that there is very low variability in our scores over different random initializations of our algorithm.



(a)

(b)

(c)

Figure 3: Error bars for internal validation metrics for K-medoids using Manhattan distance over 6 different random seeds and 4 K-values

Table 3: Error statistics of Silhouette Scores for random seed experiment

Optimal K	Clustering Algorithm	Distance Metric	Mean	Std. Deviation
5	K-medoids	Manhattan	0.032	0.0024
	K-means	Euclidean	0.0523	0.00596
	Agglomerative	Euclidean	0.019	0.0
6	K-medoids	Manhattan	0.0335	0.0015
	K-means	Euclidean	0.0465	0.0076
	Agglomerative	Euclidean	0.024	0.00
7	K-medoids	Manhattan	0.0331	0.004
	K-means	Euclidean	0.0413	0.008
	Agglomerative	Euclidean	0.023	0.0
9	K-medoids	Manhattan	0.03	0.0014
	K-means	Euclidean	0.034	0.002
	Agglomerative	Euclidean	0.0219	0.0

A.4 VISUALIZING STAGE-ONE META-SEQUENCES

After weekly usage patterns are characterized by stage-one clustering, we can look back at the full recorded history of a vehicle and label each week with the stage-one cluster assignments. This provides a way to visualize how the usage patterns develop over the course of the vehicle's lifetime. Figure 4 shows selected features from two selected vehicles within our dataset. Each week is highlighted by a color associated with the stage-one cluster assignment. The full meta-sequence is highlighted by these periodic assignments of clusters and can be represented by a string of cluster assignments. Over the long term course of the vehicle's lifetime, each vehicles meta-sequence represents the unique long term usage pattern.

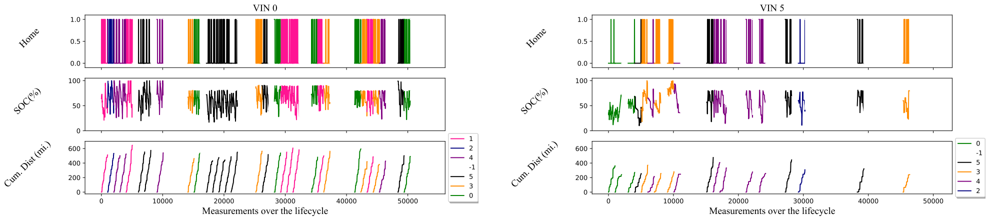


Figure 4: characterizing each vehicle's lifetime with stage-1 clustering meta-sequences

B COMPARATIVE EXPERIMENTS FOR CASCADED CLUSTERING

While we observe that K-means with Euclidean distance gives slightly better validation metrics than K-medoids with Manhattan distance, the usage profiles are qualitatively more distinct for K-medoids than for K-means. The usage profiles for K-means with Euclidean distance is shown in Figure 5. Additionally, the load profiles for the open source factory dataset are given in 6

Results using dimensionality reduction (UMAP) and then applying K-means clustering are shown in Figure 7

The heatmaps and histograms for all 6 stage 2 clusters generated using our LSTM-GCN approach are given in Figures 8 and 9. While the some of the heatnmaps may appear similar at first, the histrograms reveal differences, epecially in the proportion of weekly clusters 1, 3, and 4 present in each second-stage cluster.

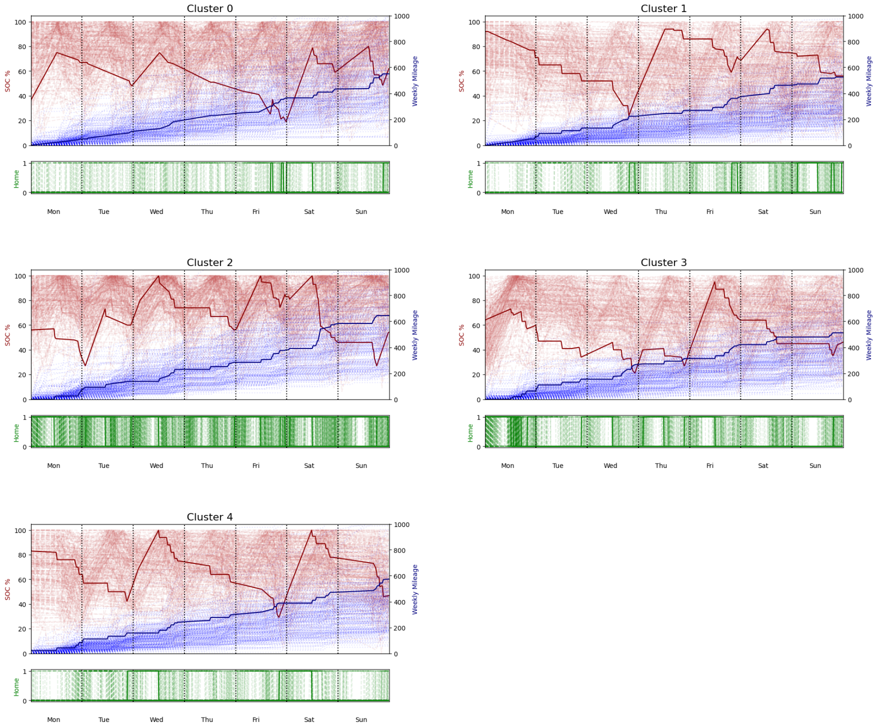


Figure 5: Weekly load profiles of Clustering Level1 clusters for SOC %

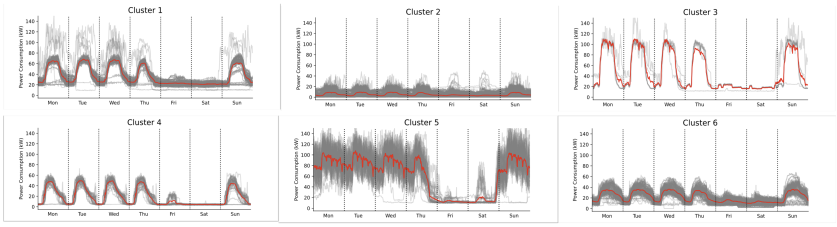


Figure 6: Weekly load profiles of Clustering Stage 1 clusters for Power

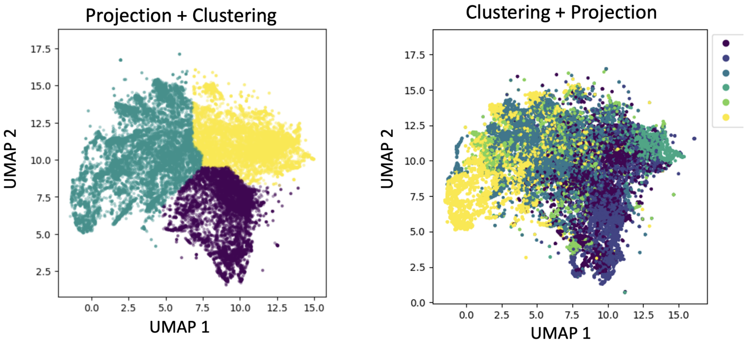
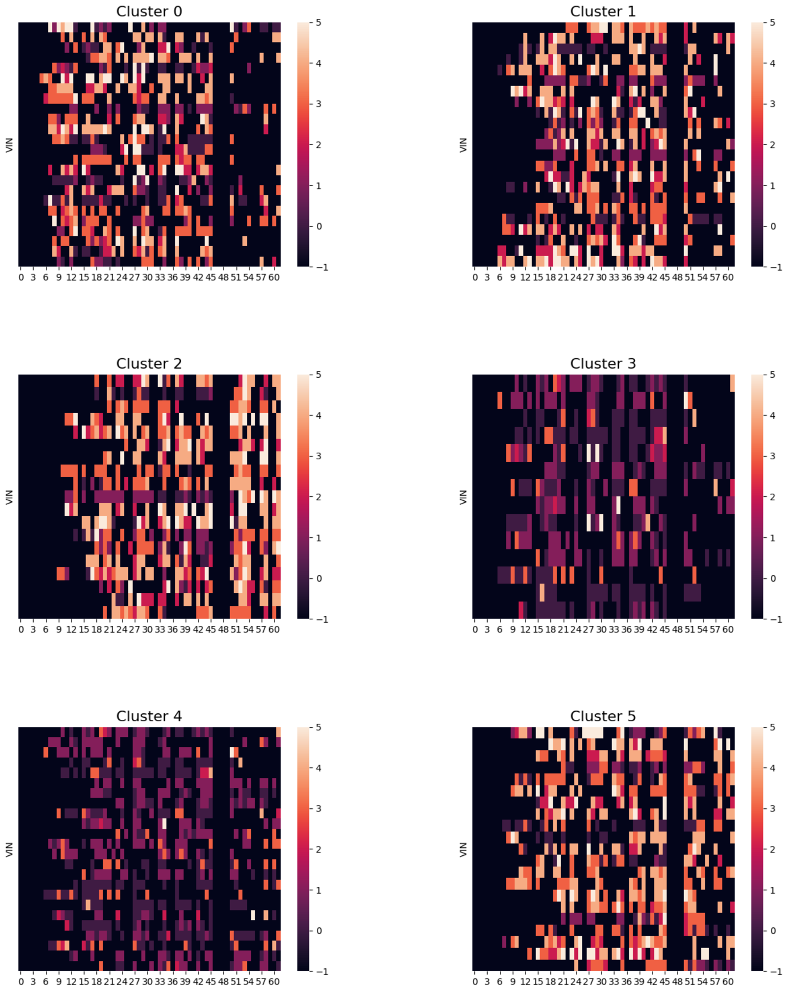


Figure 7: UMAP visualization of clustering using dimensionality reduction on the left versus raw data on the right.


Cluster 0

Cluster 1

Cluster 2

Cluster 3

Cluster 4

Cluster 5

Figure 8: Level-2 cluster distribution heatmaps

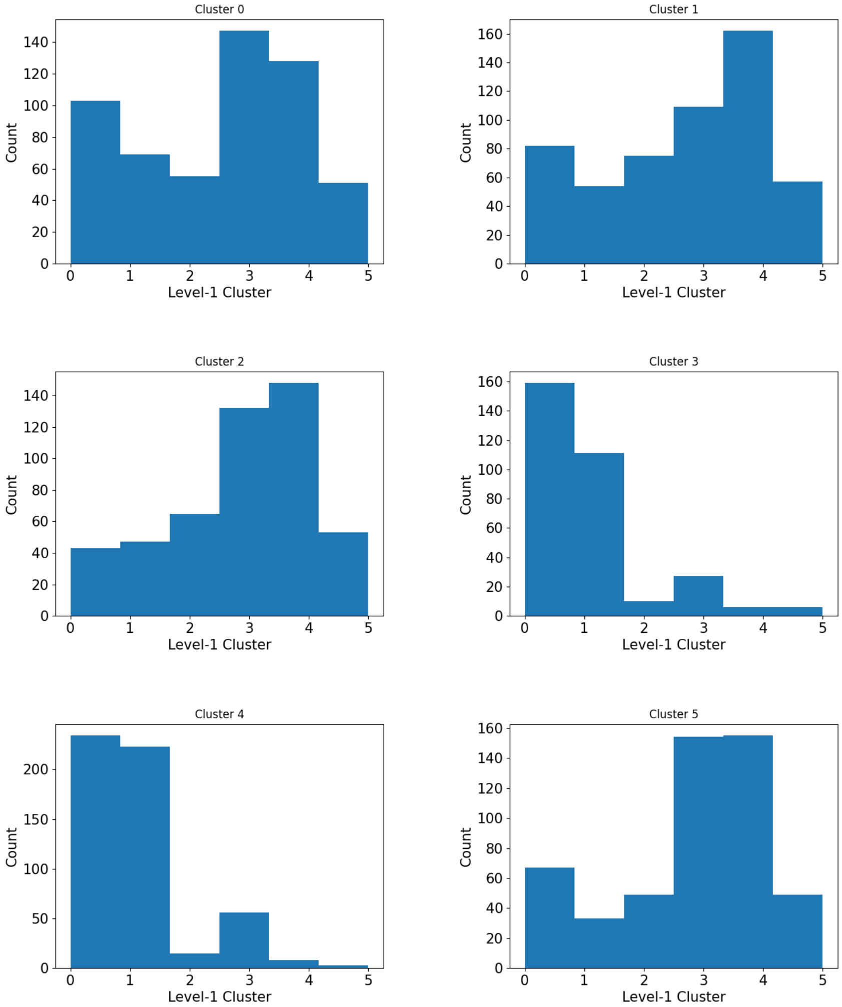


Figure 9: Level-2 cluster distribution heatmaps