Constraint-Guided Clustering for Identifying in-Vehicle Electronic Control Units from Voltage Data

Bogdan Groza[1], **Patricia Iosif**[1], **Lucian Popa**[1]

[1]Faculty of Automation and Computers, Politehnica University of Timișoara
bogdan.groza@upt.ro, patricia.iosif@student.upt.ro, lucian.popa@aut.upt.ro

Abstract

Identifying in-vehicle electronic control units based on voltage characteristics has been the subject of extensive research in cybersecurity. However, the results reported so far generally depend on restricted datasets and supervised learning. In this work, we show that clustering, i.e., unsupervised learning, of voltage characteristics, is in fact more challenging when done on a larger pool of electronic control units as several out-of-the-box clustering methods and metrics will fail to determine the correct number of clusters when exerted over a large dataset. To overcome this issue, we propose a new methodology that takes advantage of domain-specific constraints, which guide the search toward the correct number of electronic control units in a car, or even in a larger pool of units from several cars. We introduce two new metrics: correctness, which measures the success ratio with respect to the constraints, and divergence, which measures the consistency of the clustering, and show that they provide a strong indication for the optimal number of clusters. In this specific context, both metrics prove to be more reliable than the widely used Silhouette score, Davies-Bouldin and Calinski-Harabasz indexes. We successfully test our methodology on the largest dataset available today for in-vehicle voltage characteristics and discover new insights regarding the number of devices.

Code — https://github.com/LucianPopaLP/ECUScan

Introduction

Modern vehicles depend on dozens of miniature computers, called Electronic Control Units (ECUs), that are responsible for a wide range of tasks, from basic engine or body control, up to advanced autonomy-related functions. Fingerprinting in-vehicle ECUs has gained lot of traction in the research community due to important applications in cybersecurity, e.g., detecting a malicious device planted on the network or proving the origin of a message. To clarify why this is achievable, Figure 1 shows 5 randomly selected bits from the dataset we use (Popa et al. 2022). The Controller Area Networks (CAN) standard mandates a differential voltage of around 2 Volts for a dominant bit, but in practice, reaching this value depends on microscopic physical characteristics and manufacturing imperfections that cannot be cloned. In

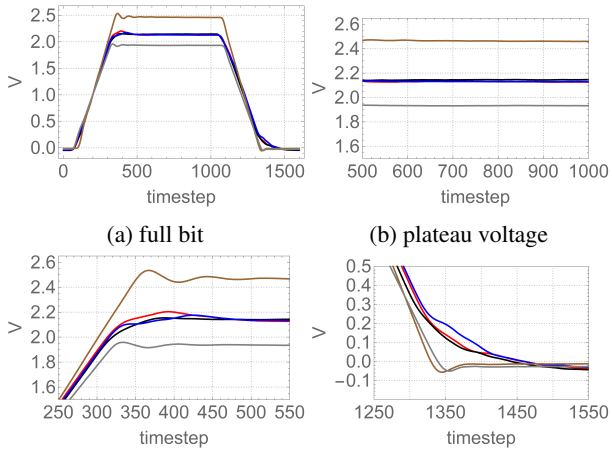


Figure 1: Voltage pattern for five randomly selected bits

plot (a) the 3 bits in the middle are slightly hard to distinguish but the 2 bits at the upper/lower edge are easy to separate. By turning to the plateau plot (b) there are already 4 voltage levels visible, while at the tip of the rising edge (c) and the bottom of the falling edge (d) the differences are much more dramatic, allowing one to easily see 5 distinctly different patterns.

Most research works on ECU fingerprinting rely on a very small pool of cars (if not just one car) and less than a dozen ECUs, although a car can have more than a hundred ECUs distributed over several buses. Moreover, most of the previous approaches rely on supervised learning. Generally, a fingerprint is learned, e.g., by using deep neural networks, and associated to the identifier (ID) of the CAN frame. The IDs and ECUs responsible for their transmission are fixed by the manufacturer and do not change, however this allocation is usually not published. Therefore, while in one car there may be dozens or even more than a hundred IDs, there are generally only a few ECUs from which these IDs originate. That is, the sender ECU can be the same for more than one ID. Clustering, i.e., determining the exact number of ECUs, can save costs associated to learning individual fingerprints for each ID, not to mention that it can be used to map the ECUs from a network by using their voltage fingerprints.

In this work, we use the largest available dataset for ECU voltage characteristics, ECUPrint (Popa et al. 2022). Rather than using supervised learning algorithms to learn the pattern of one ID or another, which is well known to work, we start from asking a more general question: how many ECUs are behind all these samples? To answer it, we need clustering and, to our surprise, the problem appears to be harder than it seems as many out-of-the-box algorithms and metrics fail. In particular, we chose three widely used metrics for determining the optimal number of clusters: Silhouette score (Rousseeuw 1987), Davies-Bouldin (Davies and Bouldin 1979) and Calinski-Harabasz indexes (Caliński and Harabasz 1974). These metrics are decades old, but their endurance is a proof of their effectiveness. Their failures suggest two things: that there may be pitfalls in the clustering from the original work (Popa et al. 2022) and that better separation scores should be defined. Figure 2 provides a partial overview of the samples from the dataset when reduced with PCA-2 (all cars included, except for the heavy-duty John Deere tractor that is too far from the center of this plot). Clearly, some samples are too close to each other and 2 features may not be enough for separation. As summarized in Table 1, our results show that in 4 of the 10 cars the number of ECUs is slightly different from (Popa et al. 2022). Our finding is corroborated by existing clustering metrics on the refined/rescaled data as well as by manual inspection of the results, online content or diagnostic information that we retrieved about the vehicles. The number of samples and IDs in Table 1 is after removing inconsistent samples and adding unclassified IDs from the original dataset (Popa et al. 2022).

We call our approach constraint-guided clustering since we rely on one domain-specific constraint which is key to the results that follow: bits that are extracted from the datafields of frames having the same ID must originate from the same ECU. This restriction comes from the CAN standard (ISO 2024): each CAN frame begins with an ID that is subject to the arbitration procedure, and thus multiple ECUs can be responsible for its bits, but only one ECU can win the arbitration and write the datafield. Therefore, the IDs of data-frames are uniquely assigned to ECUs, since otherwise more than one ECU could win the arbitration, resulting in an arbitration error. We use this domain-specific constraint to evaluate the correctness of the clustering. For this reason, one may also call our approach validation-based clustering; since we iterate until the result satisfies our need. While we later rely on spectral clustering, an additional advantage is that our methodology is agnostic and allows the use of any clustering algorithm. Briefly, our contributions can be enumerated as follows:

1. we provide a new clustering methodology where unsupervised learning is refined with domain-specific constraints, a procedure which we call *constraint-guided clustering* where the constraints guide the selection of the optimal number of clusters rather than being embedded into the clustering algorithm itself,
2. we introduce two new metrics: clustering *correctness score* and *divergence score* which are culprit in determining the optimal number of clusters, providing a good and intuitive visualization for its selection,
3. we show that, on the current dataset, our method and metrics are superior in determining the correct number of clusters when compared to classical metrics employed to determine the optimal number of clusters,
4. we test the above methodology on the largest existing dataset for voltage fingerprints, i.e., a dataset containing 57 ECUs from 10 cars, and determine that previous research has in fact missed 1–3 ECUs in 3 of them, and incorrectly identified 1 extra ECU in another, thus providing a correction for previous statistics-based methods.

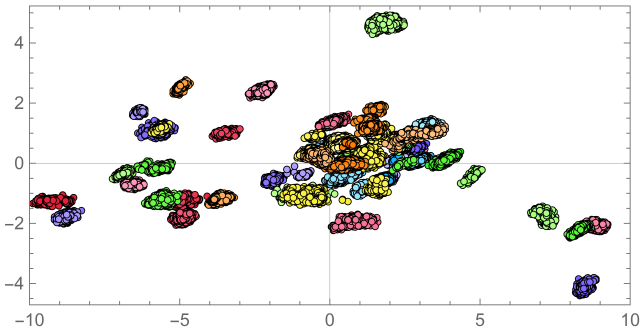


Figure 2: Partial overview of the dataset with PCA-2

Car	ECUPrint	Ours	IDs	Samples
John Deere	3	3✓	39	4,021
Dacia Duster	3	3✓	11	8,942
Opel Corsa	4	4✓	28	9,131
Ford Ecosport	4	5↑	85	20,044
Dacia Logan	6	6✓	45	31,297
Hyundai i20	7	6↓	40	17,767
Hyundai ix35	6	6✓	26	19,856
Honda Civic	6	6✓	43	14,567
Ford Fiesta	6	7↑	47	21,729
Ford Kuga	9	11↑	70	28,024

Table 1: Cars and the number of determined ECUs

Related Works

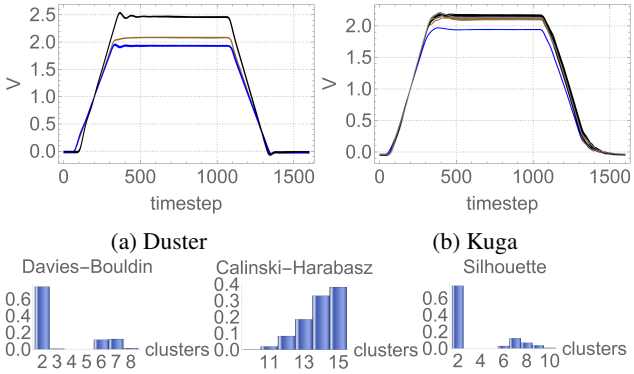
Voltage-based techniques have been commonly proposed for identifying ECUs, but clustering attempts are rarer and occurred only in a few recent works. The first attempt was done by (Kulandaivel et al. 2019), but their approach relies on clock-skews which can be cloned by adversaries (Sagong et al. 2018), making them insecure and require more time to collect – clock offsets require seconds to accumulate while a voltage fingeprint can be collected in a few micro-seconds. Regarding voltages, (Popa et al. 2022) use 4 voltage features to identify ECUs based on simple statistical distances. (Zhang and Li 2024) also use the previous dataset and, after slicing and denoising the bits, rely on DBSCAN (Ester et al. 1996) for clustering. Other attempts are based on a very small pool of cars and ECUs. For example, (Xun et al. 2023) use a combination of frame time and voltage data (Foruhandeh et al. 2019) on 17 ECUs from 2 cars. (Liu et al. 2021)

propose a clustering approach based on the Mahalanobis distance using voltage data from 2 cars with 2 and 9 ECUs. Needless to say, the experimental pool is not even close to the ECUPrint dataset, the largest publicly available, which contains 10 cars and more than 50 ECUs (Popa et al. 2022), and is also used in what follows.

Our work is in direct relation and inspired by the classic constrained-based clustering (Wagstaff et al. 2001), i.e., COP-KMeans. However, COP-KMeans tries to assign each sample to a cluster, such that the constraints are not violated, and fails if this is not possible. Our methodology is different since we run the clustering algorithm several times with no constraints on a random selection of samples and assign a score based on how well the constraints are satisfied. Conceptually, this procedure may suggest a relation with adversarial clustering and actor-critic frameworks. However, we do not use adversarial generation of realistic-looking samples as in the case of GAN-based clustering (Mukherjee et al. 2019) and thus our work is more distant from such approaches. Actor-critic approaches have been also proposed for clustering (Lee and Van Der Schaar 2020), but these belong to reinforcement learning since the actor is tuned by the feedback received from the critic. Another method based on reinforcement learning is proposed by (Dai et al. 2024). Their algorithm implements a graph consistency reward scheme based on minimizing the inter-cluster connectivity to determine the optimal number of clusters. (Huang et al. 2024) introduce a new structure-based clustering algorithm with an encoding tree that guides the clustering optimization through a better selection of clustering features. (Chen et al. 2025) improve anchor-based multi-view clustering by introducing semantic and structural constraints to ensure learned anchors are varied and uniformly distributed across all clusters. (Guo et al. 2024) propose an algorithm to optimize k-center clustering using sample-level must-link and cannot-link constraints.

Recent works explore the concept of contrastive clustering, for example (Kallidromitis et al. 2021) combines it with the Silhouette score and the Davies-Bouldin index. They introduce a novel approach using neural processes to automate the generation of augmentations by constructing multiple representations of the same function through context sampling. Another method proposed by (Li, Zhang, and Su 2023) employs contrastive and deep clustering elements for feature engineering with data-level constraints. There is also more recent progress in deep clustering which leverages the use of deep neural networks for clustering high-dimensional data. (Vardakas, Papakostas, and Likas 2024) use soft Silhouette scores which sets room for a probabilistic interpretation that can be optimized for deep clustering. (Liu et al. 2024) provide a methodology that adds the human-in-the-loop for validation. Deep graph clustering in attribute missing graphs is studied by (Tu et al. 2024).

Regarding performance indexes, (Gösgens, Tikhonov, and Prokhorenkova 2021) present a theoretical validation framework for identifying the most suitable cluster similarity indexes for various applications. The Davies–Bouldin Index is also used to assess clustering performance by (Nguyen et al. 2018), who introduce a novel approach capable of simultaneously performing clustering and classification, referred to as clustering-induced kernel learning. The Silhouette score, Davies–Bouldin and Calinski-Harabasz indexes are successfully used to evaluate the optimal number of clusters on synthetic data by (Charrad et al. 2010).



(c) optimal number of clusters reported for Kuga

Figure 3: Bits from (a) 3 ECUs in Duster, (b) 11 ECUs in Kuga and (c) clusters reported for Kuga

Interestingly, while there is an enormously rich body of works dedicated to clustering, these works generally address datasets with image, video or textual data (Zhou et al. 2024) and not automotive datasets, voltage in particular. This can be explained by the more recent emergence of this field, fewer datasets and the niche orientation in cybersecurity.

Methodology

We begin by showing how an out-of-the-box solution fails. Then we proceed to the description of the methodology proposed in this work.

Motivation: Failures of Out-of-the-Box Methods

As a motivating example, in Figure 3 (a) and (b) we show sample bits from the simplest car in the dataset, the Duster having 3 ECUs, and from the most complex car in the dataset, the Kuga with 11 ECUs. The three clusters (ECUs) in the Duster are immediately distinguished by k-means. Three well-known metrics, i.e., Davies-Bouldin, Calinski-Harabasz indexes and Silhouette score, when used for the cluster size set to $k \in [2, 16]$, correctly report the optimal number of clusters as 3. For the Kuga however, there is a strong overlap of the bits from several ECUs, which is visible on the upper side of the plot from Figure 3 (b). When running k-means, all three methods, i.e., Davies-Bouldin, Calinski-Harabasz and Silhouette, fail to identify the correct number of clusters. Moreover, the results reported are unstable along the 1000 runs as shown in the histograms from Figure 3 (c). If we take the most common of the reported values, the Silhouette score and Davies-Bouldin index report the optimal number of clusters as 2, while the Calinski-Harabasz index reports it as 15. There are 11 clusters according to our later findings (validated with all metrics), but the previous answers are far from this.

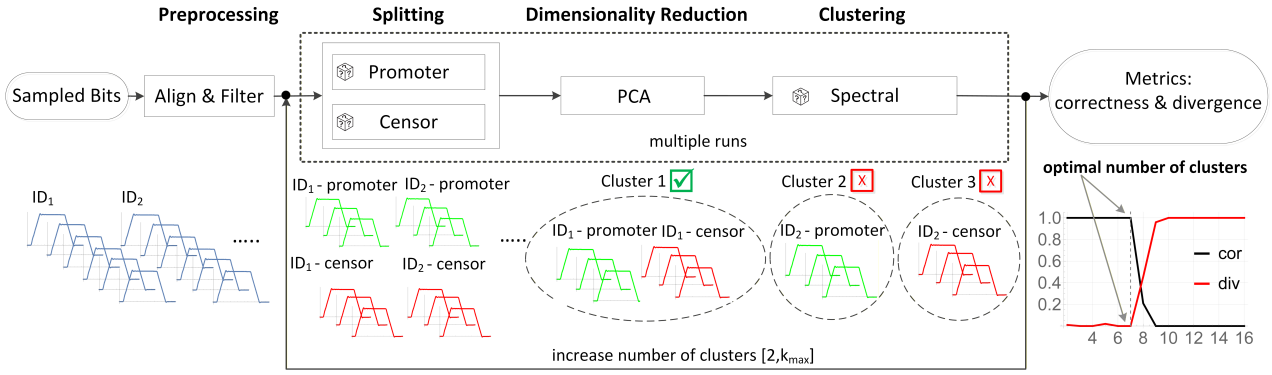


Figure 4: Overview of the clustering methodology, from pre-processing to metrics for deriving the correct number of clusters

It is of course reasonable to assume that various parameters can be tuned so that the above scores may be able to determine the right number of clusters. This kind of empirical tuning is what happens in many of the related works, where authors are tuning one model until it works for the specifics of the car they are targeting. But such an approach may not be desirable on a generic voltage dataset without prior access to the ground truth when one wants to determine the correct number of ECUs from the voltage fingerprints. Therefore, our aim here is to provide a generic methodology that works against the largest dataset available at the current time.

Concept

Figure 4 shows an overview of the proposed methodology. The dataset is first aligned and filtered. For each ID, an even number of samples is selected, if there are fewer samples than the desired number of samples, then new samples are randomly picked from the existing ones until the target number is met. The samples are randomly selected and half go into the promoter while the other half in the censor. The selected samples are concatenated for the promoter and for the censor, then parsed with PCA (Principal Component Analysis) for dimensionality reduction. The output goes to spectral clustering and the resulting clusters are evaluated based on the correctness and divergence scores. As the figure suggests, the clustering is considered valid when the promoter-censor pairs are in the same cluster and incorrect otherwise. Then the number of desired clusters is increased and the procedure run again until the maximum number of clusters is reached. We choose to work with spectral clustering due to the flexibility in choosing parameter γ which controls the width of the RBF kernel and allows for a more controlled separation of the clusters.

We assume that the clustering experiment $\mathcal{C}$ is performed n times, $n \geq 2$, for each number of clusters $i \in [2, k_{\max}]$, each time with a fresh seed and fresh promoter-censor pairs (note that in this way clustering results may vary on each run). Let $\mathsf{Valid}(i)$ be the collection of all valid outputs, that is, outputs where all promoter-censor pairs are in the same cluster when i is the target number of clusters (note that the same output/clustering may occur more than once during the n runs). And let $\mathsf{Distinct}(i)$ be the set containing all the distinct results up to reordering of the labels, valid or not, returned during the n runs when i is the target number of clusters. For the above sets, we consider that for any $i \in [2, k_{\max}]$ there are no empty clusters returned. If this happens, then, naturally, those values of i for which there are empty clusters are ignored in the computation of the optimal number of clusters (if some clusters are empty, the result doesn't correspond to the current target). We define the following two metrics:

- **correctness score** as the ratio of valid runs to the total number of runs, which is the probability of the clustering algorithm to return an output such that all promoter-censor pairs are in the same cluster (have identical labels) when the total number of clusters is set to i, that is:

$$\mathrm{cor}_i = \left|\mathsf{Valid}(i)\right| \times n^{-1}, i \in [2, k_{\max}] \quad (1)$$

- **divergence score** div_i as the ratio of distinct clusters to the number of runs, or at the extremes, it is 0 if there is a single clustering reported (up to reordering of the labels) for all of the n runs, and 1, if each of the n runs reports a different clustering (regardless of reordering the labels):

$$\mathrm{div}_i = \left(\left|\mathsf{Distinct}(i)\right| - 1\right) \times (n-1)^{-1}, i \in [2, k_{\max}] \quad (2)$$

The antagonistic relationship between these two scores is a deliberate choice. Note that the divergence score is minimized to 0 when the same cluster is returned during all of the n runs. But even if the same cluster is returned by all of the n runs, this cluster is not necessarily valid. For this reason, we also need the correctness score to check if the clustering is valid. In the ideal case, if there is a single valid cluster returned for n runs, we will have $\mathrm{cor}_i = 1.0$ and $\mathrm{div}_i = 0.0$ for at least one $i \in [2, k_{\max}]$ – this is indeed what generally happens later in the experiments.

We can now define the optimal number of clusters based on the correctness and divergence scores by getting the maximum index for which the correctness score is maximized and the divergence score is minimized, while we can also cross between the two and define the optimal number of clusters based on mixed correctness-divergence scores. Let

the following two sets of indexes for which the correctness score is maximized and divergence is minimized:

$$\arg \max_{i \in [2, k_{\max}]} (\text{cor}) = \left\{ i \mid \text{cor}_i = \max_{j \in [2, k_{\max}]} (\text{cor}_j) \right\} \quad (3)$$

$$\arg \min_{i \in [2, k_{\max}]} (\text{div}) = \left\{ i \mid \text{div}_i = \min_{j \in [2, k_{\max}]} (\text{div}_j) \right\} \quad (4)$$

Then the optimal number of clusters based on the two metrics can be defined as:

$$k^*_{\text{cor}} = \max \left[\arg \max_{i \in [2, k_{\max}]} (\text{cor}) \right] \quad (5)$$

$$k^*_{\text{div}} = \max \left[\arg \min_{i \in [2, k_{\max}]} (\text{div}) \right] \quad (6)$$

$$k^* = \max \left[\arg \max_{i \in [2, k_{\max}]} (\text{cor}) \bigcap \arg \min_{i \in [2, k_{\max}]} (\text{div}) \right] \quad (7)$$

Algorithm in Brief

The steps of the proposed methodology are also outlined in Algorithm 1. The algorithm iterates for k between 2 and `max_clusters` and, for each value of k, `no_attempts` trials are performed. During each trial, a fresh promoter-censor pair is selected for each ID, that is, the total number of samples for each ID are split in two, half are concatenated for the promoter and half are concatenated for the censor. Then the dimensionality is reduced using PCA, the value of γ is computed based on the mean Euclidean distance of the samples (more discussions on the selection of γ will follow) and spectral clustering is run. Then the correctness counter is incremented with the value returned by the `VerifyLabels` function, which checks that each two consecutive labels are identical, i.e., labels associated to promoter-censor pairs, and returns 1 if and only if this condition is met, otherwise it returns 0. The `Get` function is responsible for getting a unique ID for each clustering and this ID is appended to the `clusters` list (the number of distinct labels in this list is the number of clusters reported during the `no_attempts` runs). At the end of the while loop, the *correctness* and *divergence* scores are computed and appended to a list which is used to memorize these scores for each number of target clusters k. We note that divergence is computed using cluster names obtained by hashing the concatenation of sorted ID values, ensuring label invariance. These scores are returned and the optimal number of clusters is determined with the `argmin` and `argmax` functions.

Algorithm 1 Promoter-Censor Compute Scores

```
procedure PROMOTERCENSORCLUSTERING(params)
    scoresCor ← [], scoresDiv ← []
    for k=2 to max_clusters do
        valid = 0, clusters ← [], count = 0
        while count<no_attempts do
            features ← GetPromoterCensor(samples)
            reduced ← PCA16(features)
            γ ← 1 / (2×Mean(Δ(reduced)))
            labels ← Spectral(k, rbf, γ, seed)
            valid ← valid + VerifyLabels(labels)
            clusters.Append(Get(labels))
            count ← count + 1
        end while
        scoresCor.Append(valid / no_attempts)
        scoresDiv.Append((distinct(clusters)−1) / (no_attempts−1))
    end for
    return ScoresCor, ScoresDiv
end procedure
```

Experiments

In this section, we first argue on the choice of specific parameters, and then present the clustering results.

Parameter Selection

For spectral clustering, we use the RBF (Radial Basis Function) kernel which is widely applicable to various types of data and is known to handle non-linear data effectively. The selection of γ proved to be the critical factor for correctly identifying the clusters since it controls the RBF kernel, i.e., $A_{ij} = \exp\left(-\gamma \left\| x_i - x_j \right\|^2\right)$ and a larger γ leads to a more aggressive clustering with a larger number of clusters, while a smaller γ tends to group even more distant points leading to fewer clusters. We used the mean Euclidean distance $\mu(d_{ij})$ of all samples as reference for γ and empirical evidence suggested $\gamma \in \left[\frac{1}{4\mu(d_{ij})}, \frac{2}{\mu(d_{ij})}\right]$ as a good choice.

Clustering Results

We run the proposed methodology 100 times for each cluster size $k \in [2, 16]$ on an Intel 13900KS, 32GB RAM, without GPU acceleration. For each ID from the dataset we randomly select 500 samples; if there were fewer (many of the IDs have less), the selection is done with replacement. Then, during each run, we randomly select 100 samples in the promoter and 100 samples in the censor for each ID, i.e., a fresh promoter-censor pair, and a new random seed is generated for spectral clustering. We set $\gamma = \frac{1}{2\mu(d_{ij})}$ for all cars, except the heavy-duty vehicle, for which the bits are very similar and require a smaller value $\gamma = \frac{1}{4\mu(d_{ij})}$.

When testing the methodology for different values of γ the correctness-divergence plots were an excellent indicator for the right number of clusters. These plots are shown in Figure 5, for the rescaled data. The scores correlate perfectly, the only exception being the Duster case where the divergence, after reaching the minimum at 3 ECUs returns to the minimum at 16 ECUs. The correctness score is however 0 from 7 ECUs onward, indicating that it is not possible for this car to have more than 7 ECUs. For the rest of the cars, the results are excellent, with the peak correctness 100% and minimum divergence 0% matching exactly the expected number of clusters from the ground truth.

Table 2 compares the metrics introduced in this work with three well-known metrics, on the data without and with rescaling. For rescaling, we normalize each individual bit in

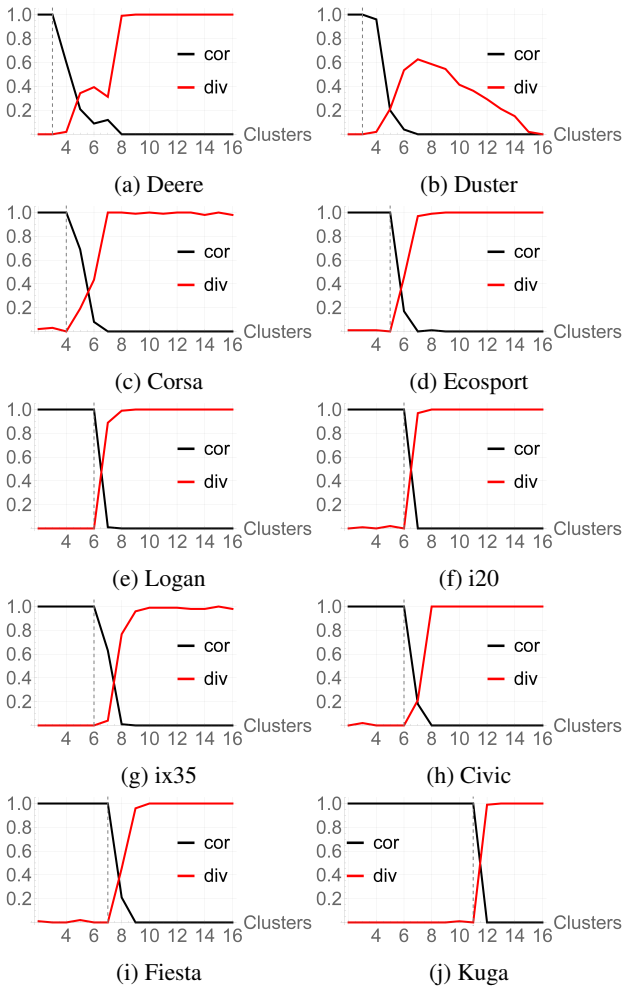


Figure 5: Correctness-divergence scores for each of the cars

the [0, 2] Volt interval, i.e., the nominal voltage according to the CAN standard. The reason for keeping the results with and without rescaling is that rescaling may suppress dominant features and reduce variance, which is undesired in cybersecurity applications like ECU identification, while, on the other hand, it is expected the rescaling will lead to more stable results. Without rescaling, the Calinski-Harabasz index is the most accurate, but still fails to give the correct answer for the Deere and the ix35. The Silhouette score works only for 5 of the 10 cars and in the other 5 cases it always shows one ECU less. The Davies-Bouldin index works only for 2 of the cars, and for one of them, i.e., the Ecosport, it oscillates between 3 and 5 (the correct number). For the rest of the cars, it can both highly overestimate (from 3 to 14 in the Duster) or underestimate the number of clusters (from 11 to 2 in the Kuga). Regarding the metrics proposed by us, they worked surprisingly well on the data without rescaling: the correctness score pointed to the right number of clusters in 9 of the 10 tests and the divergence score in 10 of the 10 tests. The only failure was for the Duster and the problem was the noise in the ID 65C samples. When we removed 3 noisy samples for this ID, the correctness score was right for all the 10 tests. All the other metrics performed poorer; Davies-Bouldin succeeded in only 1 of the 10 tests, Calinski-Harabasz in 8 of the 10 tests and Silhouette in 5 of the 10 tests. The metrics introduced in this work seem to outperform the classical ones. Regarding the results with rescaling, since the bits are now closer to each other, for a more aggressive clustering, we slightly increase γ to $\frac{1}{\mu(d_{ij})}$. We were pleasantly surprised that with rescaling our metrics work perfectly (even without the need of removing the 3 noisy files for the Duster). Moreover, the scores greatly improved for the other metrics, which now fail only on 3 cars. On the downside, both the Davies-Bouldin index and Silhouette score now fail on the Duster.

Finally, we endeavored to test the metrics on the full set of samples from all the 9 passenger cars, which gather 54 ECUs. We did not include the Deere tractor in this latter test since there are 2500 sampling points (features) for the bits collected from the Deere while for passenger cars there are only 1600. This happens because in heavy-duty vehicles the CAN bus runs at 250kbps while in passenger cars at 500kbps. Down-sampling and scaling made the Deere samples look artificial and, after dimensionality reduction, they were too far from the samples in the rest of the cars. On the data without rescaling, only the correctness and Silhouette scores reported the correct number of ECUs, i.e., 54. Divergence reported 52 clusters while Davies-Bouldin and Calinski-Harabasz indexes lead to 58 and 59 clusters respectively, which are all wrong. With rescaling, divergence correctly reported 54 clusters too and the results remained the same for correctness and Silhouette scores. Davies-Bouldin index remained at 58 clusters. The greatest failure was with the Calinski-Harabasz index (the top performing of the traditional metrics) that reported 79/80 rather than 54 ECUs. Very likely, this can be corrected by choosing a different γ but it is beyond our scope here. The value of γ, when using all passenger car samples, was set to $\gamma = \frac{2}{\mu(d_{ij})}$.

Challenges and Limitations

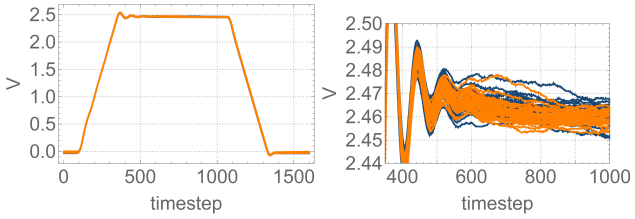
Clustering difficulty does not seem to be necessarily connected to the number of ECUs. Notably, all other metrics except ours failed on the Deere, which has only 3 ECUs, while our metrics struggle with the other car that had 3 ECUs, the Duster. The most difficult challenges that we encountered come from three harder to address causes: intra-ECU anomalies, inter-ECU similarities and inter-vehicle similarities, which are discussed next.

Intra-ECU anomalies are due to samples coming from the same ECU that are misclassified as coming from distinct ECUs due to various electrical disturbances. Interestingly, this situation was rarer than we expected, but it happened where we expect it less, in one of the simplest cars, the Duster with 3 ECUs. Figure 6 shows some anomalies in ID 65C that placed it in a different cluster. In general, filtering extreme values, using an average filter or rescaling may solve this issue. But care should be taken since this may also lead to missing some separate clusters.

Inter-ECU similarities refer to samples coming from dif-

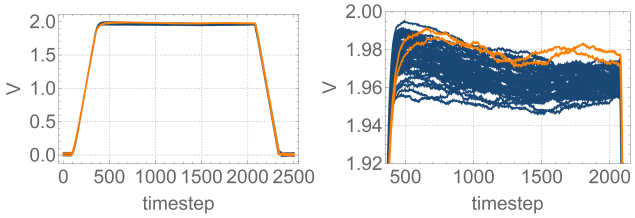
	Correctness		Divergence		Davies-Bouldin		Calinski-Harabasz		Silhouette		GT
Car	raw	scaled	raw	scaled	raw	scaled	raw	scaled	raw	scaled	
John Deere	3 ✓	3 ✓	3 ✓	3 ✓	**2 ✗**	**2 ✗**	**4 ✗**	**4 ✗**	**2 ✗**	**2 ✗**	**3**
Dacia Duster	**4 ✗**	3 ✓	3 ✓	3 ✓	**14 ✗**	**14 ✗**	3 ✓	3 ✓	3 ✓	**2 ✗**	**3**
Opel Corsa	4 ✓	4 ✓	4 ✓	4 ✓	4 ✓	4 ✓	4 ✓	4 ✓	4 ✓	4 ✓	**4**
Ford Ecosport	5 ✓	5 ✓	5 ✓	5 ✓	**3/5 ✗**	5 ✓	5 ✓	5 ✓	5 ✓	5 ✓	**5**
Dacia Logan	6 ✓	6 ✓	6 ✓	6 ✓	**4 ✗**	6 ✓	6 ✓	6 ✓	6 ✓	6 ✓	**6**
Hyundai i20	6 ✓	6 ✓	6 ✓	6 ✓	**5 ✗**	6 ✓	6 ✓	6 ✓	**5 ✗**	6 ✓	**6**
Hyundai ix35	6 ✓	6 ✓	6 ✓	6 ✓	**5 ✗**	6 ✓	**5 ✗**	6 ✓	**5 ✗**	6 ✓	**6**
Honda Civic	6 ✓	6 ✓	6 ✓	6 ✓	**5 ✗**	6 ✓	6 ✓	6 ✓	**5 ✗**	6 ✓	**6**
Ford Fiesta	7 ✓	7 ✓	7 ✓	7 ✓	**6 ✗**	**6 ✗**	7 ✓	7 ✓	**6 ✗**	**6 ✗**	**7**
Ford Kuga	11 ✓	11 ✓	11 ✓	11 ✓	**2 ✗**	11 ✓	11 ✓	11 ✓	11 ✓	11 ✓	**11**

Table 2: Optimal number of clusters for Promoter-Censor Pairs using PCA-16 without (left) and with (right) rescaling



(a) full bit (b) plateau voltage

Figure 6: Example of intra-ECU anomalies: Duster IDs 1F9 and 65C belong to the same ECU, but some samples have higher deviations, likely due to electrical disturbances

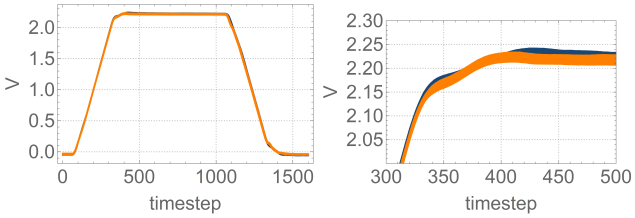


(a) full bit (b) plateau voltage

Figure 7: Example of inter-ECU similarities: Deere IDs 0CFE4421 and 1CEBFF00 belong to different ECUs, but they exhibit somewhat similar voltage patterns

ferent ECUs of the same car that are clustered together due to a similar voltage pattern. One such occurrence was in the Deere, where ID 1CEBFF00 was misclassified as belonging to another ECU, e.g., along with ID 0CFE4421 in some of the runs. We know that this cannot be so because the IDs from heavy-duty vehicles contain the source addresses (the last byte is the source address and thus 1CEBFF00 and 0CFE4421 come from two different ECUs, having addresses 00 and 21 respectively), and thus data from ID 1CEBFF00 has to be clustered together with the other frames having address 00. Figure 7 shows the full bit and details the plateau for IDs 1CEBFF00 and 0CFE4421, the similarities are obvious. A more aggressive threshold may be the solution.

Inter-vehicle similarities refer to samples clustered together as coming from the same ECU, although they come from different cars and thus they cannot belong to the same ECU. We encounter a single instance for this case where the samples from one ECU in the Ecosport seem nearly identical to the samples from another ECU in the Fiesta. Figure 8 shows the full bit similarity and the tip of the rising edge, where some differences are visible, for Ecosport ID 07E and Fiesta ID 455. This issue occurred in some runs on the full data set, but it was still possible to distinguish between the two ECUs with the proper parametrization and rescaling.



(a) full bit (b) tip of rising edge

Figure 8: Example of inter-vehicle similarity: Ecosport ID 07E and Fiesta ID 455 have very similar voltage patterns, though they do not originate from the same car

Conclusion

Briefly speaking, spectral clustering with PCA-16 and rescaling was able to perfectly cluster the ECUs in each of the 10 cars individually and even in the larger pool of nine passenger cars totaling 54 ECUs. Proper selection of parameter γ seemed to be the most critical factor for correct clustering. Empirical evidence suggested that the best value is around the inverse of the mean Euclidean distance. The metrics presented in this work, correctness and divergence, proved superior in determining the optimal number of clusters. Guided by domain-specific constraints they lead to much better results showing only a minor flaw when the data was not rescaled and a 100% accurate result on the rescaled data. This does not mean that the other metrics are worse in a general sense, it is just the fact that our metrics are designed to account for domain specific constraints and thus they work better on our problem.

References

Caliński, T.; and Harabasz, J. 1974. A dendrite method for cluster analysis. *Communications in Statistics-theory and Methods*, 3(1): 1–27.

Charrad, M.; Lechevallier, Y.; Ahmed, M. B.; and Saporta, G. 2010. On the Number of Clusters in Block Clustering Algorithms. In *FLAIRS*.

Chen, Y.; Wang, H.; Peng, J.; and Wang, Y. 2025. Anchor Learning with Potential Cluster Constraints for Multi-view Clustering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, 15939–15947.

Dai, H.; Liu, Y.; Su, P.; Cai, H.; Huang, S.; and Lv, J. 2024. Multi-View Clustering by Inter-cluster Connectivity Guided Reward. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 9846–9855. PMLR.

Davies, D. L.; and Bouldin, D. W. 1979. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence*, PAMI-1(2): 224–227.

Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X.; et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD'96: Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, 34, 226–231.

Foruhandeh, M.; Man, Y.; Gerdes, R.; Li, M.; and Chantem, T. 2019. SIMPLE: Single-frame based physical layer identification for intrusion detection and prevention on in-vehicle networks. In *Proceedings of the 35th annual computer security applications conference*, 229–244.

Gösgens, M. M.; Tikhonov, A.; and Prokhorenkova, L. 2021. Systematic analysis of cluster similarity indices: How to validate validation measures. In *International Conference on Machine Learning*, 3799–3808. PMLR.

Guo, L.; Jia, C.; Liao, K.; Lu, Z.; and Xue, M. 2024. Efficient constrained K-center clustering with background knowledge. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 20709–20717.

Huang, J.; Feng, Q.; Wang, J.; Huang, Z.; Xu, J.; and Wang, J. 2024. SEC: more accurate clustering algorithm via structural entropy. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 12583–12590.

ISO. 2024. ISO11898-1. Road vehicles — Controller Area Network (CAN) — Part 1: Data link layer and physical coding sublayer. Standard, 3rd edition, International Organization for Standardization.

Kallidromitis, K.; Gudovskiy, D.; Kazuki, K.; Iku, O.; and Rigazio, L. 2021. Contrastive neural processes for self-supervised learning. In *Asian Conference on Machine Learning*, 594–609. PMLR.

Kulandaivel, S.; Goyal, T.; Agrawal, A. K.; and Sekar, V. 2019. CANvas: Fast and inexpensive automotive network mapping. In *28th USENIX Security Symposium (USENIX Security 19)*, 389–405.

Lee, C.; and Van Der Schaar, M. 2020. Temporal phenotyping using deep predictive clustering of disease progression. In *International conference on machine learning*, 5767–5777. PMLR.

Li, H.; Zhang, L.; and Su, K. 2023. Dual mutual information constraints for discriminative clustering. In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, 8571–8579.

Liu, H.; Hu, P.; Zhang, C.; Li, Y.; and Peng, X. 2024. Interactive deep clustering via value mining. *Advances in Neural Information Processing Systems*, 37: 42369–42387.

Liu, N.; Moreno, C.; Dunne, M.; and Fischmeister, S. 2021. vProfile: Voltage-based anomaly detection in controller area networks. In *2021 Design, Automation & Test in Europe Conference & Exhibition (DATE)*, 1142–1147. IEEE.

Mukherjee, S.; Asnani, H.; Lin, E.; and Kannan, S. 2019. Clustergan: Latent space clustering in generative adversarial networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, 4610–4617.

Nguyen, K.; Dam, N.; Le, T.; Nguyen, T. D.; and Phung, D. 2018. Clustering induced kernel learning. In *Asian Conference on Machine Learning*, 129–144. PMLR.

Popa, L.; Groza, B.; Jichici, C.; and Murvay, P.-S. 2022. ECUPrint—Physical fingerprinting electronic control units on CAN buses inside cars and SAE J1939 compliant vehicles. *IEEE Transactions on Information Forensics and Security*, 17: 1185–1200.

Rousseeuw, P. J. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20: 53–65.

Sagong, S. U.; Ying, X.; Clark, A.; Bushnell, L.; and Poovendran, R. 2018. Cloaking the clock: Emulating clock skew in controller area networks. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, 32–42. IEEE.

Tu, W.; Guan, R.; Zhou, S.; Ma, C.; Peng, X.; Cai, Z.; Liu, Z.; Cheng, J.; and Liu, X. 2024. Attribute-missing graph clustering network. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 15392–15401.

Vardakas, G.; Papakostas, I.; and Likas, A. 2024. Deep clustering using the soft silhouette score: Towards compact and well-separated clusters. *arXiv preprint arXiv:2402.00608*.

Wagstaff, K.; Cardie, C.; Rogers, S.; Schrödl, S.; et al. 2001. Constrained k-means clustering with background knowledge. In *Icml*, volume 1, 577–584.

Xun, Y.; Deng, Z.; Liu, J.; and Zhao, Y. 2023. Side channel analysis: A novel intrusion detection system based on vehicle voltage signals. *IEEE Transactions on Vehicular Technology*, 72(6): 7240–7250.

Zhang, G.; and Li, Y. 2024. Voltage inspector: Sender identification for in-vehicle CAN bus using voltage slice. *Computers & Security*, 145: 104017.

Zhou, S.; Xu, H.; Zheng, Z.; Chen, J.; Li, Z.; Bu, J.; Wu, J.; Wang, X.; Zhu, W.; and Ester, M. 2024. A comprehensive survey on deep clustering: Taxonomy, challenges, and future directions. *ACM Computing Surveys*, 57(3): 1–38.