Detecting CAN Masquerade Attacks with Signal Clustering Similarity

Pablo Moriano*, Robert A. Bridges†, Michael D. Iannacone†
*Computer Science and Mathematics Division, †Cyber Resilience and Intelligence Division
Oak Ridge National Laboratory
{moriano, bridgesra, iannaconemd}@ornl.gov

***Abstract*— Vehicular Controller Area Networks (CANs) are susceptible to cyber attacks of different levels of sophistication. Fabrication attacks are the easiest to administer—an adversary simply sends (extra) frames on a CAN—but also the easiest to detect because they disrupt frame frequency. To overcome time-based detection methods, adversaries must administer masquerade attacks by sending frames in lieu of (and therefore at the expected time of) benign frames but with malicious payloads. Research efforts have proven that CAN attacks, and masquerade attacks in particular, can affect vehicle functionality. Examples include causing unintended acceleration, deactivation of vehicle's brakes, as well as steering the vehicle. We hypothesize that masquerade attacks modify the nuanced correlations of CAN signal time series and how they cluster together. Therefore, changes in cluster assignments should indicate anomalous behavior. We confirm this hypothesis by leveraging our previously developed capability for reverse engineering CAN signals (i.e., CAN-D [Controller Area Network Decoder]) and focus on advancing the state of the art for detecting masquerade attacks by analyzing time series extracted from raw CAN frames. Specifically, we demonstrate that masquerade attacks can be detected by computing time series clustering similarity using hierarchical clustering on the vehicle's CAN signals (time series) and comparing the clustering similarity across CAN captures with and without attacks. We test our approach in a previously collected CAN dataset with masquerade attacks (i.e., the ROAD dataset) and develop a forensic tool as a proof of concept to demonstrate the potential of the proposed approach for detecting CAN masquerade attacks.**

I. Introduction

Modern vehicles are complex cyber-physical systems containing up to hundreds of electronic control units (ECUs). ECUs are embedded computers that communicate over a (few) Controller Area Networks (CANs) to help control vehicle functionality, including acceleration, braking, steering, and engine status, among others. CANs are vulnerable to cyber exploitation, both by adversaries with direct physical access (e.g., through the standard on-board diagnostic [OBD] II port) and remote access (e.g., Bluetooth, 5G). This increasing connectivity enables more advanced vehicle features at the expense of expanding the attack surface. By hijacking ECUs, attackers may stealthily manipulate CAN frames resulting in life threatening incidents. For example, malicious frame injection through cellular networks has resulted in unintended acceleration, vehicle brake deactivation, and rogue steering wheel turning [1], [2].

CAN attacks are commonly classified using a three-tiered taxonomy that includes fabrication, suspension, and masquerade attacks [3], [4]. Fabrication attacks inject extra frames, whereas suspension attacks remove benign frames; consequently, both categories usually disturb regular frame timing on the bus and can be accurately detected using time-based methods [5], [6], [7]. Masquerade attacks require the adversary to send frames in lieu of (and therefore at the expected time of) benign frames but with malicious payloads. In masquerade attacks, adversaries first suspend frames of a specific ID and then inject spoofed frames that modify the content of the frames instead of their timing patterns. Hence, masquerade attacks are the stealthiest CAN attacks.

Masquerade attacks can still be detected because they alter the regular relationships of a vehicle's subsystems. Using an example attack from the ROAD [4] dataset, an adversary that gains control of the ECU(s) that communicate the wheel speed signals (four nearly identical signals) can modify the frames to break the near perfect correlation, which will stop the vehicle (regardless of the driver's actions). By understating the regular relationships of the vehicle's CAN signals, this condition can be flagged as anomalous, even if the modified signals are not abnormal when considered individually.

The widespread dependence of modern vehicles on CANs, combined with the security vulnerabilities has been meet with a push to develop intrusion detection systems (IDSs) for CAN. Generally, there are two types of IDSs methods: signature and machine learning (ML). Signature-based methods rely on a predefined set of rules for attack conditions. Behavior that matches the expected signature is regarded as an attack [8], [9], [10]. However, given the heterogeneous nature of the CAN bus in terms of transmission rates and broadcasting, effective rules for detecting attacks are difficult to design, which contributes to high rates of false negatives [11]. In contrast, ML-based methods profile benign behavior to identify anomalies or generalized attack patterns when the traffic does not behave

Workshop on Automotive and Autonomous Vehicle Security (AutoSec) 2022
24 April 2022, San Diego, CA, USA
ISBN 1-891562-75-4
https://dx.doi.org/10.14722/autosec.2022.23028
www.ndss-symposium.org

as expected.

In doing this, many ML-based methods leverage the CAN's frame payloads [12], [13]. Note that in passenger vehicles, signals (sensor values communicated in CAN frames) are encoded into the frame payloads via proprietary (nonpublic, original equipment manufacturer-specified) mappings. Some IDSs operate on the binary payload (raw bits) [12], whereas others operate on the time series of signal values [13]. Processing the binary payload has a set of associated challenges. First, there is a semantic gap with respect to the signals encoded in the payload. This means that a single CAN frame's payload usually contains several signals encoded in different formats, including byte ordering, signedness, label and units, and scale and offset [14]. Second, detecting subtle masquerade attacks requires analyzing the payload content because the correlation between certain signals may change when the frame content is modified during an attack; hence, analyzing translated signals is a promising avenue. Thus, considering the relationship between signals is important for achieving a more effective defense against advanced masquerade attacks.

In this work, we propose a forensic framework to decide if recorded CAN traffic contains masquerade attacks. The proposed framework works at the signal-level and leverages time series clustering similarity to arrive at statistical conclusions. In doing so, we use available and readable signal-level CAN traffic in benign and attack conditions to test our framework. The results obtained from our evaluation demonstrate the capability of the proposed framework to detect masquerade attacks in previously recorded CAN traffic with high accuracy. Our contributions in this paper are summarized as follows:

- We detail a CAN forensic framework based on time series clustering similarity for detecting masquerade attacks. The proposed framework is based on (1) clustering time series using agglomerative hierarchical clustering (AHC); (2) computing a clustering similarity; and (3) performing hypothesis testing using the clustering similarity distributions to decide between benign and attack conditions.
- We perform a sensitivity analysis of detection capabilities with respect to the type of AHC used. We report our results and offer possible explanations.
- We evaluate the proposed framework on masquerade attacks from the ROAD dataset [4]. Evaluation results show very high effectiveness of detecting attacks of different levels of sophistication. Our results indicate that the proposed forensic framework can be built upon to yield a viable real-time IDS, but using these results to craft a short-time-to-detection IDS is future work.

II. Related Work

Our research is informed by past work leveraging time series signal correlations for context characterization of cyber-physical systems. Here, we provide an overview of related work in this area.

Ganesan et al. [15] introduced the notion of using pairwise correlations of vehicular sensor readings (e.g., speed, acceleration, steering) to characterize behavioral context. They used it for cluster analysis to identify distinct driver behaviors and detect potential attacks. Li et al. [16] leveraged correlations from multiple sensors to train a regression model that estimates a targeted sensor value. They used the difference between the estimated and observed sensor values as an anomaly signature. Sharma et al. [17] proposed to compute Pearson correlation matrices of geolocation-related signals (e.g., latitude, longitude, elevation, speed, heading) to estimate the state of neighboring vehicles and detect location forging misbehavior based on correlation matrices' distance. Guo et al. [18] proposed Edge Computing Based Vehicle Anomaly Detection, which focuses on analyzing the time and frequency domains of sensor data to detect anomalies. In the first step, they flag abrupt changes in the correlations of sensor readings in the time domain as an indication of anomalies. For more accurate anomaly detection in the second step, they further analyze the sudden change in sensor readings by computing the change in power spectral density (PSD) of sensor data in the frequency domain. Under anomalous circumstances, the PSD is expected to be higher in the high-frequency band. He at al. [19] explored using correlations between heterogeneous sensors to identify consistency among sensor data (e.g., acceleration, engine RPM, vehicle speed, GPS) and then utilize the data to detect anomalous sensor measurements. They accomplished this by embedding the relationship of multiple sensors into an autoencoder and pinpointing anomalies based on the magnitude of the reconstruction loss. Leslie [20] developed an unsupervised learning method to detect malicious traffic over J1939 data. This method converts categorical features to numerical features with a one-hot encoding scheme and uses an ensemble AHC algorithm that integrates multiple linkage options.

Compared with the studies mentioned above, the present paper is unique in that we model temporal and signal-wise dependencies between CAN signals using time series clustering [21]. Specifically, we use AHC to generate a hierarchical relationship between signals known as a dendrogram [22]. Using a hypothesis test, we show that masquerade attacks are detectable by the resultant distribution of clustering similarities. In addition, our method is tested on real CAN data containing hundreds of signals, as opposed to previous methods that used a dozen signals at most.

III. Methods

Our focus is on processing a set of N signals (i.e., time series, $\mathcal{S} = \{X^1, X^2, \ldots, X^N\}$) obtained from a CAN log captured during a vehicle's drive. The subsections below explain the mathematical details of each step of our method and the data source used to perform this research. The proposed framework applies AHC (see § III-A) to produce a dendrogram of clusters of $\mathcal{S}$. Given two captures, each producing its corresponding dendrogram, we compute a similarity between the dendrograms using the CluSim method (see § III-B) [23]. Finally, the pairwise similarities from each capture's dendrograms are used to create a hypothesis test to distinguish between a benign CAN capture and an attack CAN capture.

A. Hierarchical Clustering

Hierarchical clustering is a method that outputs a hierarchy of clusters (i.e., a set of nested clusters that are organized in a tree-like diagram known as dendrogram). It works by transforming a proximity matrix into a sequence of nested partitions. Figure 1 depicts the details of a hierarchical clustering and its subsequent dendrogram using the agglomerative approach. The mathematical formulation of hierarchical clustering can be found in the Appendix A. Agglomerative

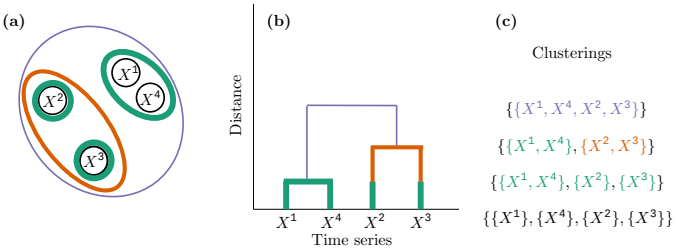

Fig. 1. A hierarchical clustering using the agglomerative approach. (a) An example of a hierarchical clustering: here, $\{X^1, X^2, X^3, X^4\}$ is a set of time series to be clustered. They are grouped in a hierarchy of clusters by color (i.e., aquamarine, orange, purple). The thickness of the clusters represents how the hierarchy is built: close time series (aquamarine), more distant time series (orange), and most distant time series (purple). (b) Corresponding dendrogram of the hierarchical clustering depicted in (a). Time series are placed in the x-axis, and their relative distance is shown in the y-axis. Cluster colors correspond with the ones in (a). (c) Different clusters at each level of the hierarchy written explicitly. Note that cutting the dendrogram horizontally creates clusterings.

algorithms[1] require a definition of dissimilarity between clusters called a *linkage*. The most popular linkages are the (a) single linkage, which is the smallest dissimilarity between two points in opposite clusters; (b) complete linkage, which is the largest dissimilarity between two points in opposite clusters; (c) average linkage, which is the average dissimilarity over all points in opposite groups; and (d) Ward's linkage, which focuses on how the sum of squares will increase when opposite groups are merged (or on the analysis of cluster variance). Ward's linkage tends to produce similar clusters as the k-means method [21].

Given a CAN capture that has been translated into its constituent signal time series, $\mathcal{S} = \{X^1, X^2, \ldots, X^N\}$, we wish to cluster these time series to produce a dendrogram that represents their hierarchical structure. Each linkage choice (a)–(d) produces a potentially different dendrogram. Understanding the most effective choice is a research question we address.

B. Clustering Similarity

Given two hierarchical clusterings (dendrograms) of a set $\mathcal{S}$, a clustering similarity quantifies a distance between them. We computed the similarity between dendrograms using the open-source CluSim method [25]. The similarity value provided by this method exists in the range $[0, 1]$, where 0 implies maximally dissimilar clusters, and 1 corresponds to identical clusterings. We parametrized the clustering similarity method by letting $r = -5.0$ and $\alpha = 0.9$. Figure 2 shows a comparison between similarity scores of three dendrograms. The key advantage of CluSim is that it does not suffer from critical biases found in previous methods (e.g., normalized mutual information) and works for hierarchical clusterings, including in conditions of skew cluster sizes and a different number of clusters. We detail the main steps of CluSim in the Appendix B.

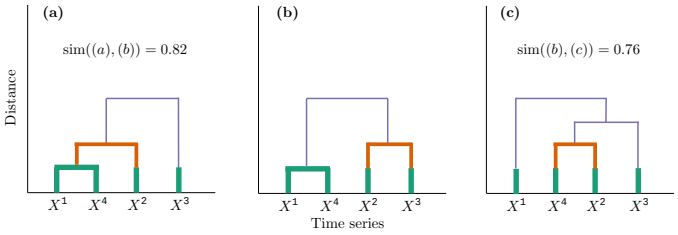

Fig. 2. Hierarchical clustering similarity comparison using the CluSim method [23]. Both (a) and (c) depict slightly modified dendrograms from that of (b). The similarity between (a) and (b) dendrograms is 0.82, whereas the similarity between (b) and (c) dendrograms is 0.76. This reflects that (a) and (b) are more similar than (b) and (c).

C. Dataset

We used the ROAD dataset [4] to test our proposed forensic framework. The ROAD dataset is an open set of CAN data collected from a real vehicle with fabrication attacks and a few advanced attacks (e.g., masquerade attacks). All of these attacks are physically verified (i.e., the effect of the CAN manipulation is observed and documented). Notably, masquerade attacks are also included but are simulated from the targeted ID fabrication attacks by removing the benign frames of the target ID. The ROAD dataset provides translated CAN time series following a similar schema used by Hanselmann et al. [13]. The fundamental advantage of using the ROAD dataset over prior released datasets is that it contains realistic, verified, and labeled attacks as opposed to synthetic ones. This opens the possibility for the evaluation, comparison, and validation of CAN signal-based IDS methods in realistic conditions.

We tested our forensic framework on the subset of masquerade attacks within the ROAD dataset. Each masquerade attack file in the ROAD dataset contains time series from hundreds of IDs that have a few to dozens of signals each. Table I shows the files we used from the ROAD dataset. Specifically, we tested the following attacks (in increasing order of detection difficulty): correlated signal, max speedometer, max engine coolant temperature, reverse light on, and reverse light off. In the correlated signal attack, the correlation of the four wheel speed values is altered by manipulating their individual values. In the max speedometer and max engine coolant attacks, the speedometer and coolant temperature values are modified to their maximum. In the reverse light attacks, the state of the reverse lights is altered to not match what gear the car is using

[1]There are two main categories of hierarchical clustering: agglomerative and divisive. Agglomerative places each object in its own cluster and gradually merges these atomic clusters into larger ones until all objects belong to a single cluster. Divisive reverses the process starting with all objects belonging to a single cluster and dividing them into smaller pieces [24]. Here, we used the agglomerative approach by virtue of its simplicity.

TABLE I
CAN CAPTURES USED FROM THE ROAD DATASET [4].

	Description	# Files	Used	Duration (min)
Training	Dynamometer Various Ambient	10	✓	108.2
	Road Various Ambient	2	✓	70.6
	Total	12	12	178.8
Testing	Correlated Signal Fabrication Attack	3	✗	1.3
	Correlated Signal Masquerade Attack	3	✓	1.3
	Fuzzing Fabrication Attack	3	✗	0.7
	Max Engine Coolant Temp Fabrication Attack	1	✗	0.4
	Max Engine Coolant Temp Masquerade Attack	1	✓	0.4
	Max Speedometer Fabrication Attack	3	✗	3.9
	Max Speedometer Masquerade Attack	3	✓	3.9
	Reverse Light Off Fabrication Attack	3	✗	2.1
	Reverse Light Off Masquerade Attack	3	✓	2.1
	Reverse Light On Fabrication Attack	3	✗	3.2
	Reverse Light On Masquerade Attack	3	✓	3.2
	Accelerator Attack (In Drive)	2	✗	2.7
	Accelerator Attack (In Reverse)	2	✗	3.2
	Total	33	13	10.9

(i.e., the reverse light is on when the vehicle is not in reverse, and the reverse light is off when the vehicle is in reverse).

We used the complete set of 12 files to characterize the behavior in benign conditions (≈3 hours of data). We used each of the files in the masquerade attack category: 3 files in the correlated attack (≈1.3 minutes), 3 files for the max speedometer attack (≈3.9 minutes), 1 file for the max engine coolant attack (≈0.4 minutes), 3 files for the reverse light on attack (≈3.2 minutes), 3 files for the reverse light off attack (≈2.1 minutes) to characterize the attack conditions.

D. Pipeline Detailed Steps

The following steps are performed to arrive at the results.

1) Same Length Time Series Transformation: Each ID has a characteristic frequency that is unique in most cases. We modified the time series to have the same frequency by linearly interpolating them in common timestamps. We chose a baseline frequency of 10 Hz because it is the lowest frequency in the IDs in this dataset. This ensures that $\forall X^i \in \mathcal{S},\ |X^i| = T$. Time series of the same length enable easier computation of similarity. After this, we also discarded any constant time series and normalized each remaining series to the unit norm.

2) Time Series Correlation Computation: We computed pairwise Pearson correlations [26] among time series. Time series that have a positive correlation are expected to move in tandem (i.e., when one measurement increases or decreases, the other measurement also increases or decreases). Pearson correlation values that are close to ± 1.0 indicate strong positive or negative correlation. As vehicle subsystems are dependent, we expect (1) clusters of correlated signals (e.g., increasing speed of the vehicle matches increases in the speedometer reading and the speed of all four wheels), and (2) such relationships to be broken or significantly changed upon a cyber attack.

3) Hierarchical Clustering Computation: These pairwise correlations populate a correlation matrix, which is used as the input for AHC. The output is a dendrogram depicting hierarchies between clusters. We explored the effect of linkage selection (i.e., single, complete, average, Ward) in our detection framework.

4) Similarity Distribution Computation: Once each dendrogram has been computed for each file, we compute empirical distributions of similarity between pairs of dendrograms using the method described in § III-B. We focus on two distinct groups. The first group is composed of all dendrograms derived from files in benign conditions (i.e., 12 files). In doing so, we computed pairwise similarities of dendrograms in this group, that is $\binom{12}{2} = 66$ possible combinations. The second group comes from the similarity between dendrograms in each category of attack (i.e., correlated, max speedometer, max engine coolant, reverse light on, reverse light off) and each of the files in benign conditions. This produces a varying number of combinations based on the number of files in each of the attack categories.

5) Hypothesis Testing: We used the Mann-Whitney U test [27] and set the significance level to 0.05 to test the null hypothesis that the distribution underlying benign conditions is the same as the distribution underlying attack conditions. The Mann-Whitney U test is a nonparametric test often used as a test of difference in location between distributions.

E. Motivational Preliminary Data Analysis

As a first step to investigate our hypothesis that masquerade attacks will disrupt clustering based on correlation of the CAN signals, we compute and visualize the CluSim similarity (§ III-B) between every pair of files in the dataset discussed above (12 benign files and 13 masquerade attack files of five different attack scenarios, all in their signal time series format). More specifically, we follow the steps describe in § III-D1 to interpolate time series to identical time steps, § III-D2 to compute Pearson correlation for each pair of signals in a CAN file, and § III-D3 to produce a dendrogram for the signals in each file. We then apply CluSim, and visualize the pairwise similarity results in Figure 3.

To see if the benign files signal cluster dendrograms do indeed "look" similar to each other but different than signal cluster dendrograms from masquerade attacks, we apply AHC to all the files based on their CluSim similarities to each other. Figure 3 shows the resulting dendrogram revealing four main clusters. Notably the first two (from left to right) contain all but one benign file (11/12), and only two (of 13) attack files. This provides strong empirical motivation to pursue a more formal detection experiment based on hierarchical clustering of signals.

IV. RESULTS

Here we present our results on the efficacy of the proposed forensic framework for detecting masquerade attacks in the CAN bus. We focus on analyzing the detection capabilities for each of the attacks described in § III-C. Figure 4 plots the probability density functions in the correlated attack (in benign and attack conditions) using the Gaussian kernel density estimate implementation from `seaborn` [28] with a default bandwidth. We study the effect of the linkage selection (in the hierarchical clustering) for distinguishing between benign and attack conditions: (a) single, (b) complete, (c) average, and

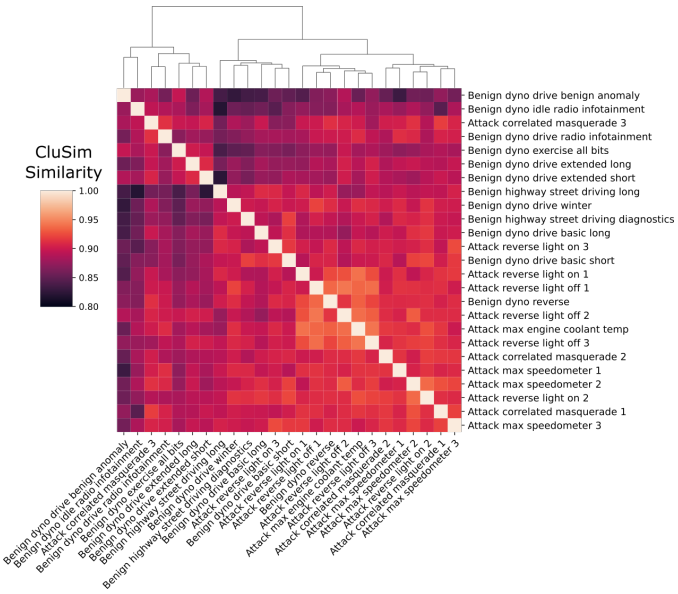

Fig. 3. CluSim cluster similarity heatmap for each pair of files from the ROAD dataset (12 benign files, 13 masquerade attack files of five attack scenario types) depicted. For each file, a hierarchical clustering dendrogram is produced based on similarity of the file's CAN signals. For each pair of files, CluSim produces a similarity measure between the two file's dendrograms using Ward's linkage, $r = -5.0$, and $\alpha = 0.9$, which is visualized by colors in the heatmap. Atop the heatmap is a dendrogram showing hierarchical clustering of all files based on their the CluSim similarities. We used Euclidean distance on these similarities with Ward linkage. Notably, there are four main clusters. From left to right, the first and second contain all benign files except `Benign dyno reverse` file, and only two attack files, i.e., `Attack correlated masquerade 3` and `Attack reverse light on 3`, while the final two clusters contain all remaining attack files and the aforementioned benign file. As a preliminary analysis, this heatmap and clustering give positive results for masquerade attack detection by comparing Pearson correlation signal clusters.

(d) Ward. We also report the p-value, using three decimals, of the associated Mann-Whitney U test to compare the two distributions in the inset; statistically significant values (i.e., p-value < 0.05) are printed in bold. Recall that we fixed the scaling parameter $r = -5$ for comparing hierarchical clusterings. This is because we want to capture differences at higher levels of the dendrograms, in which the focus is on coarser groups of multiple correlated signals, instead of more fine-grained groupings of individual to a few signals, in which not much emphasis is on their correlations.

Overall, we find that detecting attacks depends heavily on (1) the linkage function used to compute the hierarchical clusterings and (2) the severity of the attack in terms of the number of correlations perturbed. Specifically, out of the five attacks studied, the method based on Ward's linkage detected all of them (5 of 5), followed by complete linkage (4 of 5). Both single and average linkage methods detected fewer attacks (3 of 5). We report the p-values resulting from running the forensic framework for the remaining attacks (i.e., max speedometer, max engine coolant, reverse light on, and reverse light off) for each linkage in Table II. We elaborate on each attack scenario below.

TABLE II
STATISTICAL HYPOTHESIS TEST RESULTS (p-VALUES).

Attack Scenario	Single	Complete	Average	Ward
Correlated	**0.005**	**0.002**	0.123	**0.000**
Max Speedometer	**0.003**	**0.017**	**0.007**	**0.000**
Max Engine Coolant	0.251	0.065	**0.006**	**0.008**
Reverse Light On	0.378	**0.007**	0.057	**0.004**
Reverse Light Off	**0.039**	**0.004**	**0.004**	**0.000**

Statistically significant values are printed in bold.

A. Correlated Attack

Figure 4 shows the comparison of similarity distributions in the correlated attack. Among these, we found that the framework that used the average linkage (i.e., [c]) is not able to differentiate between benign and attack conditions. We also noticed that the Ward's method has the most distinctive difference (i.e., smaller p-value).

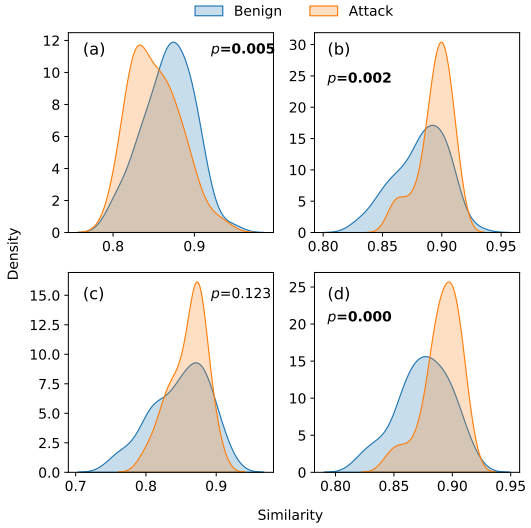

Fig. 4. Empirical distribution comparison of the correlated attack for each linkage selection: (a) single, (b) complete, (c) average, and (d) Ward. Results from these distributions appear in the first row of Table II.

B. Max Speedometer Attack

Table II shows that for the max speedometer attack, each linkage option produces statistically significant differences. We notice again that the Ward linkage produces the most distinctive results. We believe that speedometer readings correlate closely with wheel speed and engine readings, so when the speedometer value is manipulated (via attack) to appear maximally, correlations broken with these signals should be captured by the similarity distributions.

C. Max Engine Coolant Attack

Table II shows the results of the max engine coolant temperature attack. We notice that only average and Ward linkages detect significant differences. In this attack, the engine coolant signal value is set to maximum, which may cause correlations with other engine signals to differ.

D. Reverse Light On Attack

Table II shows the comparison of similarity distributions in the reverse light on attack. Note that only the complete and Ward linkages produce statistically significant differences (i.e., [b] and [d]). We also note that, although statistically significant, these p-values are not as small as in the correlated attack, which is a consequence of having an attack that is more difficult to detect. This suggests that fewer correlated signals are affected under this attack, (i.e., only a binary [1 bit] signal was targeted).

E. Reverse Light Off Attack

Table II shows that for the reverse light off attack, each linkage method produces statistically significant differences. Among these, Ward's linkage produces the most significant difference, followed by average and complete linkages. The single linkage produces the least significant result, but it still meets the threshold.

F. Detection Evaluation

We compute and compare the performance of the proposed framework for classifying benign and attack files. Recall that we use 12 benign files. To do so, we implement a cross-validation as follows: We set apart three benign files to be used for testing purposes (along with all attack files) and use the remaining nine files for training, that is, for computing the similarity distribution from benign files. We chose to hold out three benign files for testing to be consistent with the maximum number of attack files found in the attack dataset (i.e., correlated, max speedometer, reverse light on, and reverse light off attacks each have three attack files). We implement the above train-test split of our benign files for each of the $\binom{12}{9} = 220$ possible combinations. This experimental design allows us to decide if the difference between similarity distributions in benign and attack scenarios is statistically significant and further count the number of true positives (TP), false positives (FP), false negatives (FN), and true negatives (TN).

We use the best set of parameter values derived from our previous experiments, i.e., Ward linkage, $r = -5.0$, $\alpha = 0.9$, and a significance level for the statistical hypothesis test of 0.05. We report the following micro-averaged classification metrics based on these numbers: *Precision*, defined as $\frac{TP}{TP+FP}$, gives the likelihood that the computed similarity distribution difference can be attributed to an attack; *Recall*, defined as $\frac{TP}{TP+FN}$, gives the likelihood that attack files are detected. Since higher precision often comes at the price of lower recall (and vice versa), it is important to consider a balance of both metrics, and the standard balanced metric is the F1 score, defined as $2 \times \frac{precision \quad recall}{precision+recall}$. Table III summarizes these findings.

Because our method is unsupervised, the training set is defined by the benign files in a given fold not the attack files; hence, the false positive counts and rate are independent of the attack scenarios. In our experiment, the overall FPR is 13.64%, with per-file FPR depicted in Figure 5.

TABLE III
CLASSIFICATION RESULTS (%).

Attack Scenario	Precision	Recall	F1 score
Correlated	88.00	100.00	93.62
Max Speedometer	88.00	100.00	93.62
Max Engine Coolant	87.18	92.73	89.87
Reverse Light On	87.23	93.18	90.11
Reverse Light Off	88.00	100.00	93.62

False positive rate (FPR), defined as $\frac{FP}{FP+TN}$ equals to 13.64%. Because this method is unsupervised, the training set is defined by the benign files in a given fold not the attack files; hence, the false positive counts and rate are independent of the attack scenarios.

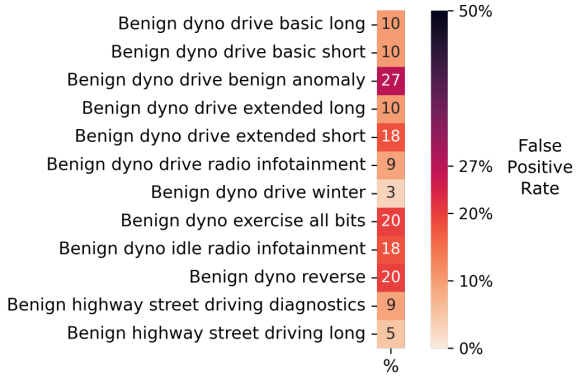

Fig. 5. False positive rates for each benign file. All are at or below 20%, except `Benign dyno drive benign anomaly` at 27%. Comparing with Figure 3, we see the lone benign file to cluster with attack files in preliminary analysis (i.e., `Benign dyno reverse`) has FPR = 20%.

Unlike the false positive and true negative counts, the true positive and false negative results will vary across different attack scenarios. For correlated, max speedometer, and reverse light off attacks, our results are identical. In these attack scenarios, recall is 100.00% meaning that all the attack configurations are detected even when changing the set of benign files used in training. In these attack scenarios, precision is 88% meaning that from the detected configurations, there are a few that come from the set of benign files or false positives. We obtain different results for the max engine coolant and reverse light on attacks. In particular, the lower results for the max engine coolant attack suggests that this attack was more difficult to detect when varying the set of benign files.

V. DISCUSSION

This work proposes a statistical forensic framework to detect masquerade attacks in the CAN bus. We quantify the empirical distribution of similarities of time series captures in benign and attack conditions. To accomplish this, we cluster time series using AHC and compute the similarity between their corresponding dendrograms. We find that masquerade attacks can be detected effectively using the proposed framework, and its discriminatory power depends on the linkage function being used in the AHC as well as the the impact of the attacks on correlated signals.

These results suggest that the proposed framework is a viable approach for detecting masquerade attacks in a forensic setting. We assume that the time series signal translation (or at least a high-fidelity translation) is readily available for use. This seems feasible with current and upcoming work in reverse engineering CAN bus signals, such as CAN-D [14].

The proposed framework detects all masquerade attacks in the ROAD dataset when the Ward linkage is used. Note that Ward's linkage (d) is an appropriate choice in this context because it tends to produce dense-enough clusters and enables the capture of meaningful changes in clustering assignations when attacks occur. In contrast, for the single linkage (a), clusters of signals tend to be spread out and often not compact enough with clusters having disparate elements. In the complete linkage (b), clusters of signals tend to be compact, but not far enough apart, with clusters having similar members. Additionally, for the average linkage (c), clusters tend to be relatively compact and relatively far apart, which strikes a balance between single and complete linkages.

We note that the detection performance may also depend on specific attack features. Here, the detection difficulty is based on the potential number of correlated signals that are affected by the attack. Thus, an attack scenario in which wheel speed signals are modified, such as in the correlated attack, has a more noticeable effect of disrupting correlation with other signals than an attack that modifies the reverse lights because the wheel speed correlation attack manipulates four highly correlated signals (and seemingly strong correlations to many other signals), whereas the reverse light attacks modify a single signal that has correlation with gear selection but not many other signals.

Detection metrics are also affected by the number of files used to compute the similarity distribution. It other words, augmenting the number of files to estimate the similarity distribution helps to have better defined distributions that are later used for comparison purposes. This explains the lower results on the max engine coolant attack that contains a single file.

To the best of our knowledge, the results from this research are the first to show systemic evidence of a forensic framework successfully detecting masquerade attacks based on time series clustering using a dataset of realistic and verified masquerade attacks. The following are some limitations of our work.

ROAD dataset conditions. The ROAD dataset was collected on a single vehicle while being exercised mostly on a dynamometer. We acknowledge that more comprehensive data collection using different vehicles may be necessary to generalize our findings. We also are aware that driving conditions may affect correlations of CAN signals, and the dynamometer conditions may be restrictive.

Parameter tuning. The proposed framework allows for flexible election of linkage functions (e.g., single, complete, average, Ward) for computing the hierarchical clusterings and the scaling parameter r and α to control the influence of hierarchical clusterings with shared lineages. Here, we fixed the values of r and α to focus on differences at higher levels of the dendrograms or in groups of correlated signals. However, we acknowledge that the optimal selection of these parameters may depend on the type of attack and driving conditions. We did not explore those variables in this research.

Not real-time detection. As is currently presented, this is not a real-time detector.

Baseline comparison. We did not compare our proposed forensic framework with other methods.

VI. Conclusion

In this research, we proposed a forensics framework for the detection of masquerade attacks in the CAN bus. To ascertain this fact in experiments, we compute time series clustering similarity. We show that the similarity of time series clusters under benign conditions exhibits statistically significant differences from the the similarity of time series clusters under attack conditions. We demonstrated these differences under different attack scenarios with different levels of sophistication using data from the ROAD dataset. This work shows that it is possible to detect masquerade attacks by effectively using the time series clustering representation of signals in the CAN bus and appropriate choices of parameters to group them.

Future work in this area includes the development of a real-time IDS that uses the principles described in this work. Additional work includes the translation of such developments to edge computing devices that can be integrated with real-world vehicle conditions.

VII. Acknowledgments

This research was sponsored in part by Oak Ridge National Laboratory's (ORNL's) Laboratory Directed Research and Development Program. This research used resources of the Compute and Data Environment for Science (CADES) at ORNL, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

References

[1] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," *Black Hat USA*, vol. 2015, p. 91, 2015.

[2] ——, "CAN message injection: OG dynamite edition," *Tech. Rep.*, 2016.

[3] K.-T. Cho and K. G. Shin, "Fingerprinting electronic control units for vehicle intrusion detection," in *Proceedings of the 25th USENIX Security Symposium*, 2016, pp. 911–927.

[4] M. E. Verma, M. D. Iannacone, R. A. Bridges, S. C. Hollifield, P. Moriano, B. Kay, and F. L. Combs, "Addressing the Lack of Comparability & Testing in CAN Intrusion Detection Research: A Comprehensive Guide to CAN IDS Data & Introduction of the ROAD Dataset," 2022, arXiv preprint arXiv:2012.14600, January 2022.

[5] H. M. Song, H. R. Kim, and H. K. Kim, "Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network," in *Proceedings of the International Conference on Information Networking (ICOIN)*, 2016, pp. 63–68.

[6] M. R. Moore, R. A. Bridges, F. L. Combs, M. S. Starr, and S. J. Prowell, "Modeling inter-signal arrival times for accurate detection of CAN bus signal injection attacks: a data-driven approach to in-vehicle intrusion detection," in *Proceedings of the 12th Annual Conference on Cyber and Information Security Research*, 2017, pp. 1–4.

[7] D. H. Blevins, P. Moriano, R. A. Bridges, M. E. Verma, M. D. Iannacone, and S. C. Hollifield, "Time-based can intrusion detection benchmark," in *Proceedings of the Workshop on Automotive and Autonomous Vehicle Security (AutoSec)*, 2021, pp. 1–6.

[8] U. E. Larson, D. K. Nilsson, and E. Jonsson, "An approach to specification-based attack detection for in-vehicle networks," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, 2008, pp. 220–225.

[9] M. Bresch and N. Salman, "Design and implementation of an intrusion detection system (ids) for in-vehicle networks," Master's thesis, University of Gothenburg, 2017.

[10] H. Olufowobi, C. Young, J. Zambreno, and G. Bloom, "SAIDuCANT: Specification-Based Automotive Intrusion Detection Using Controller Area Network (CAN) Timing," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 1484–1494, 2019.

[11] W. Wu, R. Li, G. Xie, J. An, Y. Bai, J. Zhou, and K. Li, "A survey of intrusion detection for in-vehicle networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 919–933, 2019.

[12] A. Taylor, S. Leblanc, and N. Japkowicz, "Anomaly detection in automobile control network data with long short-term memory networks," in *Proceedings of the IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, 2016, pp. 130–139.

[13] M. Hanselmann, T. Strauss, K. Dormann, and H. Ulmer, "CANet: An unsupervised intrusion detection system for high dimensional CAN bus data," *IEEE Access*, vol. 8, pp. 58 194–58 205, 2020.

[14] M. E. Verma, R. A. Bridges, J. J. Sosnowski, S. C. Hollifield, and M. D. Iannacone, "CAN-D: A Modular Four-Step Pipeline for Comprehensively Decoding Controller Area Network Data," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 10, pp. 9685–9700, 2021.

[15] A. Ganesan, J. Rao, and K. Shin, "Exploiting consistency among heterogeneous sensors for vehicle anomaly detection," SAE Technical Paper, Tech. Rep., 2017.

[16] H. Li, L. Zhao, M. Juliato, S. Ahmed, M. R. Sastry, and L. L. Yang, "Poster: Intrusion detection system for in-vehicle networks using sensor correlation and integration," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, 2017, pp. 2531–2533.

[17] P. Sharma, J. Petit, and H. Liu, "Pearson correlation analysis to detect misbehavior in vanet," in *Proceedings of the 88th Vehicular Technology Conference (VTC-Fall)*, 2018, pp. 1–5.

[18] F. Guo, Z. Wang, S. Du, H. Li, H. Zhu, Q. Pei, Z. Cao, and J. Zhao, "Detecting vehicle anomaly in the edge via sensor consistency and frequency characteristic," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5618–5628, 2019.

[19] T. He, L. Zhang, F. Kong, and A. Salekin, "Exploring inherent sensor redundancy for automotive anomaly detection," in *Proceedings of the 57th ACM/IEEE Design Automation Conference (DAC)*, 2020, pp. 1–6.

[20] N. Leslie, "An unsupervised learning approach for in-vehicle network intrusion detection," in *Proceedings of the 55th Annual Conference on Information Sciences and Systems (CISS)*, 2021, pp. 1–4.

[21] A. Javed, B. S. Lee, and D. M. Rizzo, "A benchmark study on time series clustering," *Machine Learning with Applications*, vol. 1, p. 100001, 2020.

[22] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, 2009, vol. 344.

[23] A. J. Gates, I. B. Wood, W. P. Hetrick, and Y.-Y. Ahn, "Element-centric clustering comparison unifies overlaps and hierarchy," *Scientific Reports*, vol. 9, no. 1, pp. 1–13, 2019.

[24] G. N. Lance and W. T. Williams, "A general theory of classificatory sorting strategies: 1. hierarchical systems," *The Computer Journal*, vol. 9, no. 4, pp. 373–380, 1967.

[25] A. J. Gates and Y.-Y. Ahn, "CluSim: A Python package for calculating clustering similarity," *Journal of Open Source Software*, vol. 4, no. 35, p. 1264, 2019.

[26] K. Pearson, "Notes on regression and inheritance in the case of two parents," *Proceedings of the Royal Society of London*, vol. 58, pp. 240–242, 1895.

[27] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *The Annals of Mathematical Statistics*, pp. 50–60, 1947.

[28] M. L. Waskom, "seaborn: statistical data visualization," *Journal of Open Source Software*, vol. 6, no. 60, p. 3021, 2021.

[29] T. H. Haveliwala, "Topic-sensitive PageRank: A context-sensitive ranking algorithm for web search," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 4, pp. 784–796, 2003.

APPENDIX

A. Hierarchical Clustering Definition

Here we mathematically define hierarchical clustering. A partition $\mathcal{P}$, of $\mathcal{S}$ breaks $\mathcal{S}$ into non-overlapping subsets $\{C^1, C^2, \ldots, C^m\}$, i.e., $\mathcal{S} = \bigcup_{i\in\{1,2,\ldots,m\}} C^i$. A clustering is a partition, so the elements of the partition are called clusters. A partition $\mathcal{B}$ of $\mathcal{S}$ is nested in a partition $\mathcal{A}$ of $\mathcal{S}$ if every subset of $\mathcal{B}$ is a subset of a subset of $\mathcal{A}$, i.e., $\forall C^i \in \mathcal{B}\ \exists j : C^i \subseteq C^j \in \mathcal{A}$. A hierarchical clustering is then a sequence of partitions in which each partition is nested into the next partition in the sequence.

B. Brief CluSim Overview

Here we describe how CluSim works in brevity. See Gates et al. [23] for full details. Given $\mathcal{S} = \{X^1, X^2 \ldots, X^N\}$ and a clustering $\mathcal{A} = \{C^1, C^2, \ldots, C^m\}$, first make the bipartite graph with elements of $\mathcal{S}$ on the left, clustering assignments from $\mathcal{A}$ on the right, and edges denoting containment (i.e., (X^i, C^j) is an edge if and only if X^i is in cluster C^j). Note that this can be naturally extended to a dendrogram representing a hierarchical clustering $\mathcal{A}$ by using a weighted bipartite graph, where the weight of the edges is given by a hierarchy weighting function based on the level of the cluster assignation within the hierarchical clustering. Next, the bipartite graph is projected into the $\mathcal{S}$ elements producing a weighted, directed graph that captures the inter-element relationships induced by common cluster memberships. Now equipped with a weighted, directed graph on $\mathcal{S}$, the CluSim method captures high-order co-occurrences of elements by taking into account their paths to obtain an equilibrium distribution of a personalized diffusion process on the graph, or personalized PageRank (PPR) [29], i.e., for each X^i in $\mathcal{S}$, a PageRank version with restart to X^i given by probability $1 - \alpha$ is used to produced stationary distribution $\boldsymbol{p}_i$. The element-wise similarity of an element X^i in two different clusterings $\mathcal{A}$ and $\mathcal{B}$ is found by comparing the stationary distributions $\boldsymbol{p}_i^{\mathcal{A}}$ and $\boldsymbol{p}_i^{\mathcal{B}}$ using a variation of the ℓ^1 metric for probability distributions. Finally, the similarity score of two clusterings $\mathcal{A}$, $\mathcal{B}$ is the average of element-wise similarities. CluSim is parametrized by specifying r and α. Here, r is a scaling parameter that defines the relative importance of memberships at different levels of the hierarchy. That is, the larger r, the more emphasis on comparing lower levels of the dendrogram (zoom in). In addition, α is a parameter that controls the influence of hierarchical clusterings with shared lineages. That is, the larger α, the further the process will explore from the focus data element, so more of the cluster structure is taken into account into the comparison. We used $r = 5.0$ and $\alpha = 0.9$ in Figure 2.