Locality Sensitive Hashing for Tampering Detection in Automotive Systems

Roland Bolboacă
George Emil Palade University of Medicine, Pharmacy, Science and Technology of Târgu Mureş
Târgu Mureş, Mureş, Romania
roland.bolboaca@umfst.ro

Teri Lenard
George Emil Palade University of Medicine, Pharmacy, Science and Technology of Târgu Mureş
Târgu Mureş, Mureş, Romania
teri.lenard@umfst.ro

Béla Genge
George Emil Palade University of Medicine, Pharmacy, Science and Technology of Târgu Mureş
Târgu Mureş, Mureş, Romania
bela.genge@umfst.ro

Piroska Haller
George Emil Palade University of Medicine, Pharmacy, Science and Technology of Târgu Mureş
Târgu Mureş, Mureş, Romania
piroska.haller@umfst.ro

ABSTRACT

In modern auto vehicles we find dozens of Electronic Control Units (ECUs) running several hundred MBs of code, alongside sophisticated dashboards with integrated wireless communications. While this technological advancement has brought upon a wide range of advantages and integrated features, it also exposed the modern vehicle to significant cyber threats, as documented in prior works. Unfortunately, besides traditional cyber attacks, the security and normal operation of the modern vehicle are nowadays exposed to a different kind of threat. This is the tampering, which denotes a procedure that alters the vehicle's behavior in order to gain particular advantages (e.g., financial, operational). A fundamental distinction between tampering and cyber attacks, is that tampering occurs with the owner's consent. This paper presents an approach for detecting tampering within modern vehicles. The approach leverages the advantages of sensitive hashing, namely the Exact Euclidean Locality Sensitive Hashing (E^2LSH) method. Experimental results based on a dataset collected from the On-Board Diagnostics port (OBD) of a Kia SOUL vehicle demonstrate the practical applicability of the developed methodology.

CCS CONCEPTS

• **Security and privacy → Intrusion/anomaly detection and malware mitigation**; • **Information systems → Nearest-neighbor search**.

KEYWORDS

Anomaly detection, locality sensitive hashing, controller area networks

Ares '20, August 25–28, 2020, Dublin, Ireland

ACM ISBN 978-1-4503-XXXX-X/18/06…$15.00
https://doi.org/10.1145/1122445.1122456

ACM Reference Format:
Roland Bolboacă, Teri Lenard, Béla Genge, and Piroska Haller. 2018. Locality Sensitive Hashing for Tampering Detection in Automotive Systems. In *ARES '20: The 15th International Conference on Availability, Reliability and Security, August 25–28, 2020, Dublin, Ireland*. ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/1122445.1122456

1 INTRODUCTION

The modern car has experienced profound changes in terms of its internal technological ecosystem. Nowadays, we find dozens of Electronic Control Units (ECUs) running several hundred MBs of code, alongside sophisticated dashboards with integrated wireless communications [2]. However, in the same vehicle we find the underlying communication infrastructure, which is struggling to keep up with the pace of these radical changes.

While this technological advancement has brought a wide range of advantages and integrated features, it also exposed the modern vehicle to significant cyber threats. To this end we find attackers (e.g., vehicle owners, malicious actors) exploiting software vulnerabilities (or undocumented features) in order to alter the vehicle's behavior [25, 27]. Fortunately, nowadays, we also find a wide variety of techniques coming from the scientific community [5, 17, 28], as well as from standardizing bodies [1], which aim to address the lack of built-in security mechanisms within the modern car.

Unfortunately, besides traditional cyber attacks, the security and normal operation of the modern vehicle are also exposed to a different kind of threat. This is the *tampering*, which denotes a procedure that alters the vehicle's behavior in order to gain particular advantages (e.g., financial, operational). Compared to cyber attacks, the purpose of tampering is not to cause specific damages, but to alter the system's behavior in order for the vehicle's *owner* to gain particular advantages. A fundamental distinction between tampering and cyber attacks, is that tampering occurs with the owner's consent. Furthermore, tampering usually requires physical access to the vehicle, as well as making profound changes to the vehicle's hardware (e.g., chip unsoldering, connecting hardware emulators via altered communication ports). To this end, a particular example is the heavy-duty transport vehicle domain, where frequent tampering is aimed at deactivating the NOx reduction system in order to avoid costs for repairing of parts or the costs

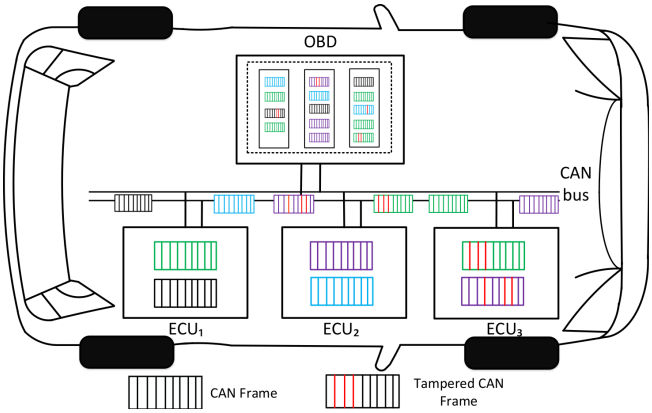

Figure 1: General architecture of the modern car with ECUs and CAN frames.

of consumables (e.g., the AdBlue reagent), which are needed for the system's correct operation. However, these changes drastically raise the amount of NOx emissions [16].

This paper presents an approach for detecting the tampering within modern vehicles. The approach leverages the advantages provided by sensitive hashing, namely the Exact Euclidean Locality Sensitive Hashing (E^2LSH) method. Introduced by P. Indyk in [9] as a method for fast searching of similar objects with a high number of characteristics, Locality Sensitive Hashing (LSH) can yield more accurate results than the mainstream techniques documented in prior works (e.g., Principle Component Analysis). Experimental results based on a dataset collected from the On-Board Diagnostics port (OBD) of a Kia SOUL vehicle [13] are used to validate the developed methodology. These also demonstrate its superior performance with respect to related techniques.

We believe that this work brings several major contributions, including: (i) it presents a new approach for detecting tampering based on the Exact Euclidean Locality Sensitive Hashing (E^2LSH), which distinguishes itself from mainstream techniques documented in prior works; and (ii) it showcases the superior performance of the developed technique with respect to techniques found in prior works in the context of a dataset captured from a real vehicle.

The remainder of this paper is structured as follows. Section 2 provides an overview of related works. This is followed by a detailed description of LSH and of the developed methodology in Section 3. Experimental results and comparison to prior works are described in Section 4. The paper concludes in Section 5.

2 BACKGROUND AND RELATED WORK

2.1 Overview of the Modern Car's Architecture

From an architectural point of view, the modern car comprises dozens of embedded devices, also known as Electronic Control Units (ECUs), which communicate with digital and/or analog sensors.

In today's modern vehicles the "backbone" communication is provided by the Controller Area Network (CAN). Standardised in 2003 [10], it is an International Standardization Organization (ISO) - defined communications bus that describes the rules for exchanging data frames between devices. Given its limitations mainly in terms of bandwidth and payload size, recently, two main improved communication infrastructures have been proposed. The CAN+ protocol was proposed by Ziermann, *et al.* in 2009 [29], and it exploits the time between transmissions to send additional data. More recently, in 2012, Robert Bosch Gmbh developed the CAN with flexible data-rate protocol (CAN-FD) [19], which brings several advantages over CAN and CAN+, amongst which the most significant being higher bandwidth and larger payload.

An overview of this architecture is visualized in Fig. 1. However, besides regular ECUs, the CAN bus can also transfer tampered frames injected by malicious actors, which can profoundly alter the behavior of ECU software. Therefore, an efficient approach is required, which can be integrated into the modern vehicle in order to detect tampering of sensitive data.

2.2 Related Work

Tampering is closely related to anomaly detection, if we consider only the scenarios that do not cause parameter deviations outside the normal functioning intervals. Therefore, in the remainder of this section, an overview of anomaly detection techniques is provided, which have been particularly developed for the automotive sector.

Groza and Murvay [6] developed an approach that focuses on the number of Control Area Network (CAN) identifiers (CIDs), their periodicity, and the entropy carried by the data-field associated to a particular CID. Based on a thorough analysis of a CAN communication trace, a significant number of constant bits have been identified. Consequently, the Hamming distance between two messages from the same sender and CID was used to define the minimum entropy. As a result, the developed approach detects replay and packet modification attacks (i.e., random changes of a packet's content).

Similarly to the work of Groza and Murvay, other researchers have acknowledged that anomaly detection algorithms need to be lightweight in order to be practically applicable to in-vehicle systems. To this end, Stabili, *et al.* [24] proposed an anomaly detection algorithm for the CAN bus based on the Hamming distance between the payloads of two consecutive CAN messages having the same identifiers. Tests consisted of randomly generated payload (fuzzing attack), as well as replay attacks. The results have shown that the approach is applicable and exhibits good performances in the case of messages with small Hamming distances. Otherwise, a large number of false positives are generated.

In the same direction of lightweight detection algorithms we mention the work of Cho and Shin [14]. Here, an anomaly detection algorithm, called Clock-based detection system, was developed. The developed algorithm was analyzed and later validated in the context of three distinct classes of attacks: new data fabrication, suspension, and masquerade. The approach essentially records the message arrival timestamps, and exploits the periodic transmission time of CAN messages for fingerprinting the transmitter ECUs. Next, for each ECU, an estimation of the clock skew was computed, which was then used as a fingerprint. Finally, the Cumulative Sum (CUSUM) was integrated to detect packet injection, omission or modification attacks.

In [15], Moore, *et al.* observed that CAN communications exhibit a certain level of regularity in terms of the timing of CAN frames.

Based on the identified communication patterns, a network-based intrusion detection strategy was developed. The developed detection strategy measures the inter-signal wait times, and issues alerts in case communications deviate from the apriorilly learned patterns. As in the case of most existing works, a clean (i.e., disturbance-free) dataset was used in the learning phase. A similar approach for detecting intrusions in in-vehicle systems according to the packet inter-arrival times (for the same CID) was documented by Sung, *et al.* in [23].

Moving towards the direction of more complex algorithms, we find the work of Narayanan, *et al.* [21]. The developed approach embraces Hidden Markov Models (HMM) to learn offline the communication patterns between ECUs. In a similar direction we find the work of Theissler, *et al.* [26], where multivariate time series were recorded and analyzed from a moving (on-road) vehicle.

In contrast to the above-mentioned studies, this work presents a methodology aimed to detect tampering (or anomaly) attempts on the data layer. An important observation is that in most of the cases tampering does not affect communications, but it mainly impacts the application layer, that is, the data transported by the underlying communication system. While most prior works targeted the underlying communication system, this paper documents a methodology aimed at the application layer. Besides this, while prior works in other fields have mainly used traditional (i.e. mainstream) machine learning algorithms, this work documents a different approach, namely the Locality Sensitive Hashing (LSH) [4]. Compared to other mainstream techniques (e.g., Principle Component Analysis – PCA), LSH does not only have the ability to work with multi-dimensional data (as the case of PCA), but, as demonstrated later by the experimental results, it also exhibits a better accuracy in the case of tampering.

3 DEVELOPED TAMPERING DETECTION METHODOLOGY

At the center of the developed methodology is a technique aimed at finding nearest neighbours in a high dimensional space. The modern car contains hundreds of sensors, which can be accessed in real-time via the On-board Diagnostic system (OBD). OBD denotes the vehicle's self-diagnostic and reporting capability, which also provides a standardized access interface and protocols for accessing sensor reports. Given the high data dimensionality, the developed approach needs to build upon multi-dimensional algorithms, namely the Exact Euclidean Locality Sensitive Hashing (E^2LSH) method.

The remainder of this section describes the building blocks of E^2LSH, and its applications to tampering detection.

3.1 Locality Sensitive Hashing

Developed as an approximate nearest neighbors solution by P. Indyk and later improved in [4], Locality Sensitive Hashing (LSH) is an innovative method for solving the nearest neighbor problem with the use of hashing. It is based on the premises that hashing points that are closer together have a higher probability of collision, than the points that are farther apart. Since it was first published, LSH has been widely used in numerous fields and applications [7, 11, 12, 18, 20].

As an extension to LSH, the Euclidean Exact Locality Sensitive Hashing [3],[22] (E^2LSH) embraces the fundamental concept of LSH, with the purpose of solving the R-near neighbor problem in Euclidean space. Given a set of values V (i.e., a set of measurements captured via the OBD port) containing data points in a d-dimensional space, for a given query point q and a radius R, E^2LSH returns a list of points for which the condition $||v - q||_2 \leq R$ is satisfied, where $||v - q||_2$ is the Euclidean Distance between v and q. More specifically, E^2LSH provides a list of points close to q within the radius R with a probability of 1 - δ, where δ is the probability that a point within R distance to q is not reported.

E^2LSH inherited from LSH its p-stable distribution. A distribution $\mathcal{D}$ over $\Re$ is called p-stable, if there exists $p \geq 0$ such that for any m real numbers $u_1 \ldots u_m$, independently and identically distributed variables $X_1 \ldots X_m$ with distribution $\mathcal{D}$, the random variable $\sum_i u_i X_i$ has the same distribution as the variable $(\sum_i |u_i|^p)^{1/p} X$. Here, X is a random variable with distribution $\mathcal{D}$ [8].

According to the previously mentioned properties, the E^2LSH structure is defined as a family of functions $G = \{g_1, ..., g_L\}$. Each $g \in G$ is defined as $g(v) = (h_1(v), ..., h_k(v))$. Here, $v \in V$ is the d-dimensional measurement that is stored in the E^2LSH structure, h_i is the hash function that is applied on each v value, k denotes the number of hash functions, and L denotes the number of g functions. For each g, a hash table of size n is constructed. It should be noted that, within the E^2LSH structure, for each $g \in G$ function, the associated hash functions h_i are uniquely constructed (as defined later in this section). In other words, for each pair of functions $g, g' \in G, g \neq g'$, and the set of hash functions $\mathcal{H}$, and $\mathcal{H}'$ associated to g, and g', respectively, we have that $\mathcal{H} \cap \mathcal{H}' = \emptyset$.

In relation to each h_i, a t function is introduced to compute the hash table key for a given g:

$$t(h_1(v), ..., h_k(v)) = \left(\sum_{j=1}^{k} r_j h_j(v) \right) \mod n, \tag{1}$$

where r_j are random integers. Lastly, the hash functions h_i are defined as:

$$h_i(v) = \left\lfloor \frac{x \cdot v + b}{w} \right\rfloor. \tag{2}$$

Here, x is a vector containing elements selected randomly from a p-stable distribution, b is a real number uniformly selected from the interval $[0, w]$, and w is the number of segments in which the real line, containing the projection $x \cdot v$, is divided into.

3.2 E^2LSH Parameter Selection

The number k of hash functions h used by each g, directly influences the query running times. Therefore, choosing the optimum value for k is crucial for the performance of the algorithm. Following the method described in [3] and [22] the value for k is approximated via the following steps. First, we estimate the value of T_c, which is the time needed to construct the g functions and to retrieve the approximate near neighbours for a query point q. Next, we estimate the value of T_g, which is the time needed to compute the distances from the query point q to all of its approximate near neighbours. Given these two estimations, the best value for k is the value for which $T_c + T_g$ is minimal, thus k should be chosen as a mean optimization of all the query points.

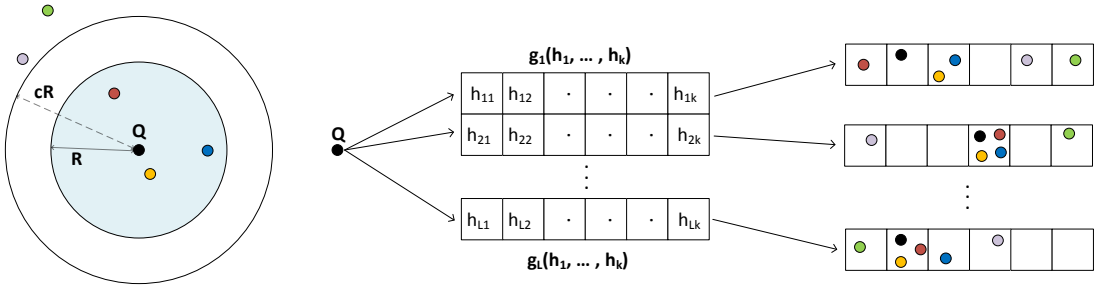

Figure 2: Simplified overview of the E^2LSH query scheme.

Based on k, the ideal number L of g functions is determined according to the assumption that a query point will collide with an existing data point with a probability of p^k. Therefore, the probability that the query point fails to collide with a near neighbor for all g_i functions is equal to $(1-p^k)^L$. As a result, a point colliding with the query point with a probability of δ, can be expressed as $1-(1-p^k)^L \geq 1-\delta$. Consequently, the best value for L is:

$$L = \left\lceil \frac{\log\left(\frac{1}{\delta}\right)}{-\log\left(1-p^k\right)} \right\rceil \tag{3}$$

Finally, the R parameter is computed in two steps. In the first step, its minimum and maximum values (i.e., R_{min} and R_{max}) are obtained by sampling data points from the tamper-free dataset, and by choosing R_{min} and R_{max} such that most of the sampled data points have the nearest neighbor in the $[R_{min}, R_{max}]$ interval. In the second step, the optimal value for $R \in [R_{min}, R_{max}]$ is chosen by querying a set of tampered points, and by selecting the best ratio between the false-positive and false-negative rates.

3.3 Tampering Detection

The developed method for tampering detection leverages the proprieties of the E^2LSH scheme, namely, the high probability that two points that are close to each other in Euclidean space hash to the same value.

At first, the tamper-free data points are pre-processed by the E^2LSH structure in order to group similar values under the same buckets in each hash table. Accordingly, each g first applies k hash functions $h_1, h_2, ..., h_k$ on each v from the dataset, resulting in a series of hash values $h_1(v), h_2(v), ..., h_k(v)$. Next, t is applied to determine the hash key within the table g.

The procedure for determining if a given query point q is tampered or not is similar to the insertion process. First, for each g, the points that collide under the same t key value, are retrieved, resulting in a list of points considered similar to q by E^2LSH. Then, to further decide if measurement q is tampered or not, for each point under the bucket identified by t, the Euclidean distance is computed with q. The point q is considered valid if at least one point exists within R distance. An overview of this procedure is shown in Fig. 2.

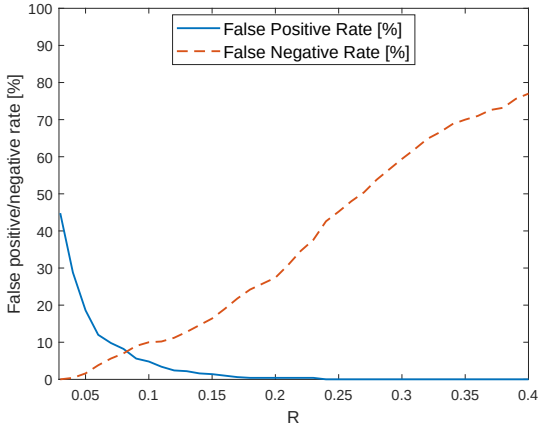

Figure 3: False Positive and False Negative Rate vs R.

4 EXPERIMENTAL ASSESSMENT

The experimental evaluation focuses on assessing the correctness and applicability of Euclidean Exact Locality Sensitive Hashing (E^2LSH) in the context of detecting data tampering in auto vehicles. Furthermore, the analysis focuses on the comparison with related detection methods.

4.1 Dataset

The data used to validate the developed approach was provided by the Hacking and Countermeasure Research Lab (HCRL) [13]. It comprises measurements collected from the On-Board Diagnostics port of a Kia SOUL vehicle. Briefly, the set consists of 94000 items, each one having 51 features, recorded over a period of 23 hours from 10 different drivers. From this collection of features, 17 features were considered relevant in terms of detecting tampering attempts, from which we mention: fuel consumption, engine speed, and engine torque (the complete set of features is summarized in Table 1). The dataset used for the construction of the E^2LSH structure, denoted by S_I, contains 7000 observations randomly selected from 8 drivers. The query dataset, denoted by S_q, contained all remaining observations from the 2 drivers. Before constructing the E^2LSH structure the dataset was normalized with respect to the limits imposed for each of the selected features.

Table 1: Description of the features used throughout the experiments.

Feature Name	Range	Unit
Fuel consumption	0-10000	mcc
Accelerator pedal value	0-100	%
Throttle position signal	0-100	%
Intake air pressure	0-255	kPA
Absolute throttle position	0-100	%
Engine speed	0-6000	rpm
Torque of friction	0-100	%
Engine coolant temperature	(-)40-(+)215	C
Engine torque	0-100	%
Calculated load value	0-100	%
Maximum indicated engine torque	0-100	%
Wheel velocity front left-hand	0-511.75	km/h
Wheel velocity rear right-hand	0-511.75	km/h
Wheel velocity front right-hand	0-511.75	km/h
Wheel velocity rear left-hand	0-511.75	km/h
Torque converter turbine speed	0-16383.75	rpm
Vehicle speed	0-200	km/h

4.2 Data Tampering

In order to generate tampered data, 2 distinct scenarios were considered. For each scenario, a subset S_t containing 500 observations were randomly selected from the query set S_q. From S_t, a total number of 17 tampered datasets were created, where gradually, for each dataset, an additional feature was tampered, starting from the 1st until the 17th feature. For each feature x_i the mean (μ_{x_i}) and standard deviation (σ_{x_i}) values were estimated. Subsequently, the tampered values were automatically computed by randomly selecting values from specific (valid) intervals in two scenarios, as follows:

- Scenario I: $(\mu_{x_i} + 3\sigma_{x_i}, \mu_{x_i} + 5\sigma_{x_i})$.
- Scenario II: $(\mu_{x_i} \pm 3\sigma_{x_i}, \mu_{x_i} \pm 5\sigma_{x_i})$.

Lastly, according to Table 1, each tampered feature was bounded to the normal parameter limits.

4.3 Parameter Computation

E^2LSH's parameters must be determined in relation to each specific dataset. Namely, we need to determine the number k of hash functions h_i specific for each g, the number L of g functions, and the radius R. By following the procedure described earlier, the following values were obtained: $k = 10$, and $L = 55$. To compute the optimal value of R, the interval margins R_{min} and R_{max} were computed by sampling random data-points from the dataset S_l. Subsequently, the optimal value for $R \in [R_{min}, R_{max}]$ was calculated using MATLAB, as shown in Fig. 3. The best result obtained for R was of 0.08. Lastly, δ, the probability that a near neighbor is not found, was set to 10%, meaning that the algorithm would report the near neighbors with a 90% probability of success.

4.4 Experimental Results

In order for the reader to understand the impact of tampering on the vehicle's parameters, three samples, each one consisting of three features are visualized in Fig. 4, 7, 10. Here, we observe the correlation between the: fuel consumption, torque of friction, and the engine's coolant temperature (Fig. 4); engine torque, load value, and engine torque (Fig. 7); and accelerator pedal, intake air pressure, and torque of friction (Fig. 10). Next, we visualize the same three features in the case of the two scenarios including tampering. Accordingly, Fig. 5, 6, 8, 9, 11, 12 illustrate the same three features in the case of Scenario I and II. Observe that in these two scenarios the features exhibit clear changes from the original dataset. Nevertheless, the parameter intervals are not exceeded. In each figure previously mentioned, Fig. 4-12, the plotted points were normalized with respect to the limits imposed by each feature (see Table 1).

Next, we proceed to the analysis of the accuracy of the developed methodology in terms of false positives and false negatives. The analysis is also performed in the context of the popular technique including Principle Component Analysis (PCA) alongside the use of Gaussian Mixture Models (GMM). The later case was used by prior studies [8] in the classification of anomalous behavior.

As mentioned earlier, the rate of false positives (FPR) and negatives (FNR) is closely linked to the value of parameter R. To this end, Fig. 3 shows the evolution of FPR and FNR. Since the value of R was set to 0.08, the recorded FPR is of 8.2% and of FNR is of 7.0%. As for PCA+GMM the recorded FPR is of 59.4%. Moving forward, Fig. 13 and 14 showcase the rate of false negatives in comparison to PCA and GMM. PCA provides a means to reduce the data dimensions, while GMM is the clustering technique used to detect anomalous behavior. As shown in these figures, we observe that PCA+GMM leads to a significantly higher level of false negatives in both experimental scenarios. Obviously, by increasing the number of features that are affected by tampering, the accuracy of both techniques improves. Nevertheless, the developed approach behaves significantly better than PCA+GMM. This is owed to the fact that, by reducing the dimensions, PCA also eliminates the slight parameter changes that are critical for the detection of tampering.

5 CONCLUSIONS

This paper approached a new threat to automotive systems, namely *tampering*, and presented a new approach for the detection of tampering. The approach leverages the advantages provided by sensitive hashing, namely the Exact Euclidean Locality Sensitive Hashing (E^2LSH) method. By using E^2LSH, the developed methodology encapsulates the ability to tackle multi-dimensional data, to model normal system behavior, and to detect abnormal (i.e., tampered) measurements. In terms of experimental results, the dataset collected from the On-Board Diagnostics port of a Kia SOUL vehicle [13] demonstrated the applicability of the approach, and its superior performance when compared to mainstream techniques such as the ones documented in related works (e.g., Principle Component Analysis, and Gaussian Mixture Models). As future work, we intend to further refine the developed technique and to implement a prototype within an auto vehicle.

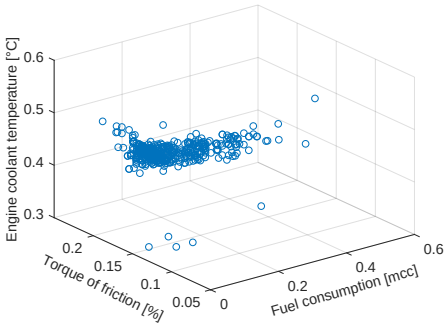

Figure 4: Tamper-free dataset (engine coolant).

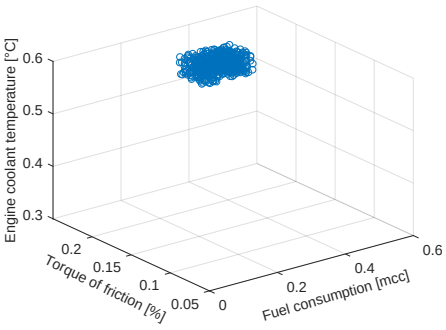

Figure 5: Scenario I tampered dataset.

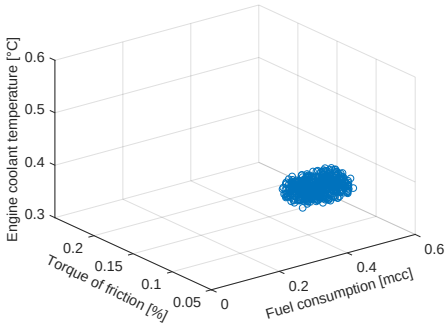

Figure 6: Scenario II tampered dataset.

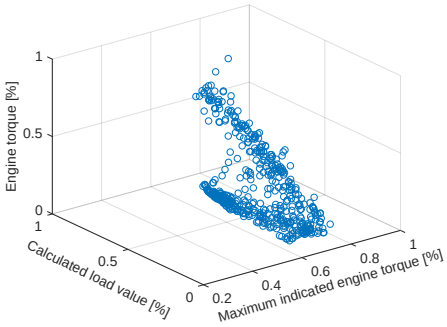

Figure 7: Tamper-free dataset (engine torque).

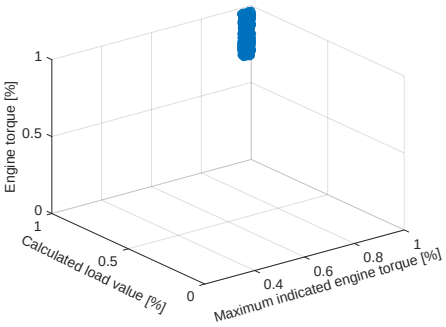

Figure 8: Scenario I tampered dataset.

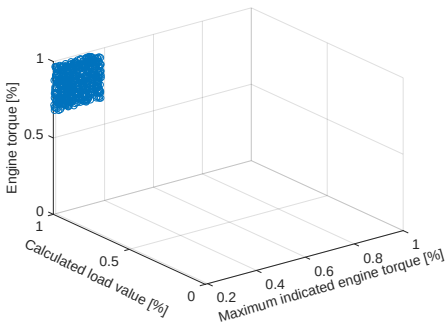

Figure 9: Scenario II tampered dataset.

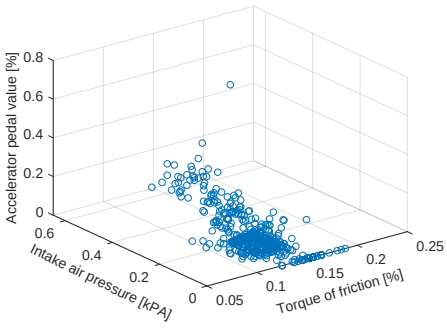

Figure 10: Tamper-free dataset (accelerator pedal).

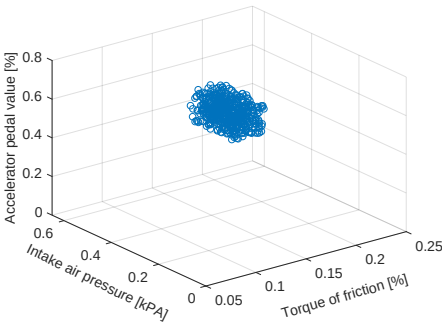

Figure 11: Scenario I tampered dataset.

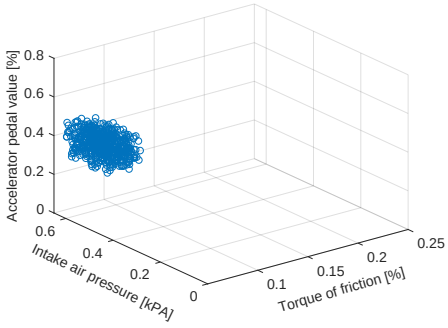

Figure 12: Scenario II tampered dataset.

6 ACKNOWLEDGMENTS

This work was funded by the European Union's Horizon 2020 Research and Innovation Programme through DIAS project (*https://dias-project.com/*) under Grant Agreement No. 814951. This document reflects only the author's view and the Agency is not responsible for any use that may be made of the information it contains.

REFERENCES

[1] AUTOSAR. 2017. Specification of Secure Onboard Communication AUTOSAR CP Release 4.3.1. *AUTOSAR* (2017).

[2] Riccardo Coppola and Maurizio Morisio. 2016. Connected Car: Technologies, Issues, Future Trends. *ACM Comput. Surv.* 49, 3, Article 46 (Oct. 2016), 36 pages. https://doi.org/10.1145/2971482

[3] Mayur Datar, Nicole Immorlica, Piotr Indyk, and Vahab S. Mirrokni. 2004. Locality-Sensitive Hashing Scheme Based on p-Stable Distributions. In *Proceedings of the Twentieth Annual Symposium on Computational Geometry* (Brooklyn, New York, USA) *(SCG '04)*. Association for Computing Machinery, New York, NY, USA, 253–262. https://doi.org/10.1145/997817.997857

[4] Aristides Gionis, Piotr Indyk, and Rajeev Motwani. 1999. Similarity Search in High Dimensions via Hashing. In *Proceedings of the 25th International Conference on Very Large Data Bases (VLDB '99)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 518–529.

[5] B. Groza and P. Murvay. 2018. Security Solutions for the Controller Area Network: Bringing Authentication to In-Vehicle Networks. *IEEE Vehicular Technology Magazine* 13, 1 (March 2018), 40–47. https://doi.org/10.1109/MVT.2017.2736344

[6] B. Groza and P. Murvay. 2019. Efficient Intrusion Detection With Bloom Filtering in Controller Area Networks. *IEEE Transactions on Information Forensics and*

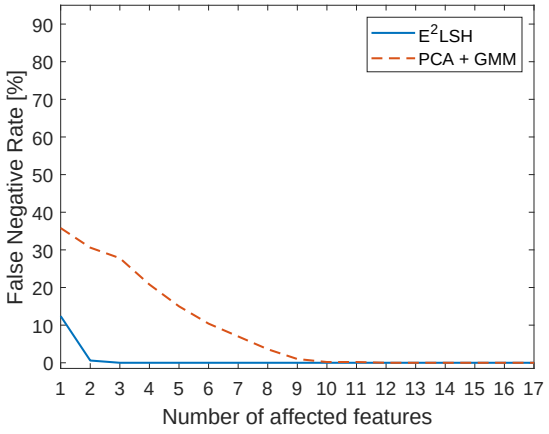

Figure 13: Accuracy of E^2LSH in comparison with PCA+GMM for Scenario I.

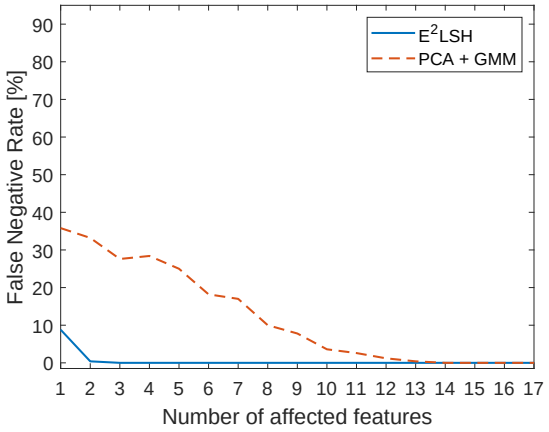

Figure 14: Accuracy of E^2LSH in comparison with PCA+GMM for Scenario II.

Security 14, 4 (April 2019), 1037–1051. https://doi.org/10.1109/TIFS.2018.2869351

[7] Parisa Haghani, Sebastian Michel, and Karl Aberer. 2009. Distributed Similarity Search in High Dimensions Using Locality Sensitive Hashing. In *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology* (Saint Petersburg, Russia) *(EDBT '09)*. Association for Computing Machinery, New York, NY, USA, 744–755. https://doi.org/10.1145/1516360.1516446

[8] P. Indyk. 2000. Stable distributions, pseudorandom generators, embeddings and data stream computation. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*. 189–197. https://doi.org/10.1109/SFCS.2000.892082

[9] Piotr Indyk and Rajeev Motwani. 1998. Approximate Nearest Neighbors: Towards Removing the Curse of Dimensionality. In *Proceedings of the Thirtieth Annual ACM Symposium on Theory of Computing* (Dallas, Texas, USA) *(STOC '98)*. Association for Computing Machinery, New York, NY, USA, 604–613. https://doi.org/10.1145/276698.276876

[10] ISO. 2003. ISO 11898-1:2003 - Road vehicles - Controller area network (CAN) - Part 1: Data link layer and physical signalling. *International Organization for Standardization* (2003).

[11] Hisashi Koga, Tetsuo Ishibashi, and Toshinori Watanabe. 2004. Fast Hierarchical Clustering Algorithm Using Locality-Sensitive Hashing. In *Discovery Science*, Einoshin Suzuki and Setsuo Arikawa (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 114–128.

[12] B. Kulis and K. Grauman. 2009. Kernelized locality-sensitive hashing for scalable image search. In *2009 IEEE 12th International Conference on Computer Vision*. 2130–2137. https://doi.org/10.1109/ICCV.2009.5459466

[13] Byung Il Kwak, Jiyoung Woo, and Huy Kang Kim. 2016. Know your master: Driver Profiling-based Anti-theft method. In *PST 2016*.

[14] Kyong-Tak Cho and Kang G. Shin. 2016. Fingerprinting Electronic Control Units for Vehicle Intrusion Detection. In *25th USENIX Security Symposium (USENIX Security 16)*. USENIX Association, Austin, TX, 911–927. https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/cho

[15] Michael R. Moore, Robert A. Bridges, Frank L. Combs, Michael S. Starr, and Stacy J. Prowell. 2017. 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* (Oak Ridge, Tennessee, USA) *(CISRC '17)*. Association for Computing Machinery, New York, NY, USA, Article 11, 4 pages. https://doi.org/10.1145/3064814.3064816

[16] Denis Pöhler, Tim Adler, Chsristopher Krufczik, Martin Horbanski, Johannes Lampel, and Ulrich Platt. 2017. Real Driving NOx Emissions of European Trucks and Detection of Manipulated Emission Systems. In *EGU General Assembly Conference Abstracts (EGU General Assembly Conference Abstracts)*. 13991.

[17] Andreea-Ina Radu and Flavio D. Garcia. 2016. LeiA: A Lightweight Authentication Protocol for CAN. In *Computer Security – ESORICS 2016*, Ioannis Askoxylakis, Sotiris Ioannidis, Sokratis Katsikas, and Catherine Meadows (Eds.). Springer International Publishing, Cham, 283–300.

[18] Zeehasham Rasheed, Huzefa Rangwala, and Daniel Barbara. 2013. 16S rRNA metagenome clustering and diversity estimation using locality sensitive hashing. *BMC systems biology* 7 Suppl 4 (10 2013), S11. https://doi.org/10.1186/1752-0509-7-S4-S11

[19] Robert Bosch Gmbh. 2012. CAN with flexible data-rate. *Vector CANtech, Inc., MI, USA, Specification Version 1.0* (2012).

[20] M. Ryynanen and A. Klapuri. 2008. Query by humming of midi and audio using locality sensitive hashing. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2249–2252. https://doi.org/10.1109/ICASSP.2008.4518093

[21] Sandeep Nair Narayanan, Sudip Mittal, and Anupam Joshi. 2016. OBD SecureAlert: An Anomaly Detection System for Vehicles. In *IEEE Workshop on Smart Service Systems (SmartSys 2016)*.

[22] G. Shakhnarovich, T. Darrell, and P. Indyk. 2006. *Locality-Sensitive Hashing Using Stable Distributions*. MITP, 61–72. https://ieeexplore.ieee.org/document/6282722

[23] H. M. Song, H. R. Kim, and H. K. Kim. 2016. Intrusion detection system based on the analysis of time intervals of CAN messages for in-vehicle network. In *2016 International Conference on Information Networking (ICOIN)*. 63–68. https://doi.org/10.1109/ICOIN.2016.7427089

[24] Dario Stabili, Mirco Marchetti, and Michele Colajanni. 2017. Detecting attacks to internal vehicle networks through Hamming distance. 1–6. https://doi.org/10.23919/AEIT.2017.8240550

[25] Y. Takefuji. 2018. Connected Vehicle Security Vulnerabilities [Commentary]. *IEEE Technology and Society Magazine* 37, 1 (March 2018), 15–18. https://doi.org/10.1109/MTS.2018.2795093

[26] Andreas Theissler. 2017. Detecting known and unknown faults in automotive systems using ensemble-based anomaly detection. *Knowledge-Based Systems* 123 (2017), 163 – 173. https://doi.org/10.1016/j.knosys.2017.02.023

[27] C. Urquhart, X. Bellekens, C. Tachtatzis, R. Atkinson, H. Hindy, and A. Seeam. 2019. Cyber-Security Internals of a Skoda Octavia vRS: A Hands on Approach. *IEEE Access* 7 (2019), 146057–146069. https://doi.org/10.1109/ACCESS.2019.2943837

[28] A. Van Herrewege, D. Singelee, and I. Verbauwhede. 2011. CANAuth - A Simple, Backward Compatible Broadcast Authentication Protocol for CAN bus. In *ECRYPT Workshop on Lightweight Cryptography 2011 (ECRYPT '11)*. 1–7.

[29] T. Ziermann, S. Wildermann, and J. Teich. 2009. CAN+: A new backward-compatible Controller Area Network (CAN) protocol with up to 16× higher data rates.. In *2009 Design, Automation Test in Europe Conference Exhibition*. 1088–1093. https://doi.org/10.1109/DATE.2009.5090826