A Framework for Consistent and Repeatable Controller Area Network IDS Evaluation

Paul Agbaje*, Afia Anjum*, Arkajyoti Mitra*, Gedare Bloom†, Habeeb Olufowobi*
*University of Texas at Arlington, Arlington, TX, USA
†University of Colorado Colorado Springs, Colorado Springs, CO, USA
Email: {pauloluwatowoju.agbaje, afia.anjum, arkajyoti.mitra, habeeb.olufowobi}@uta.edu, gbloom@uccs.edu

***Abstract*—The landscape of automotive vehicle attack surfaces continues to grow, and vulnerabilities in the controller area network (CAN) expose vehicles to cyber-physical risks and attacks that can endanger the safety of passengers and pedestrians. Intrusion detection systems (IDS) for CAN have emerged as a key mitigation approach for these risks, but uniform methods to compare proposed IDS techniques are lacking. In this paper, we present a framework for comparative performance analysis of state-of-the-art IDSs for CAN bus to provide a consistent methodology to evaluate and assess proposed approaches. This framework relies on previously published datasets comprising message logs recorded from a real vehicle CAN bus coupled with traditional classifier performance metrics to reduce the discrepancies that arise when comparing IDS approaches from disparate sources.**

I. Introduction

Modern cars can contain tens to hundreds of electronic control units (ECUs) that communicate with each other over in-vehicle networks including the controller area network (CAN) for enhanced performance, safety, and comfort. CAN is a robust serial shared bus communication standard that aims to be simple and efficient while delivering real-time and fault-tolerant performance in harsh environments. A major concern is that the CAN bus implements no security mechanism and the increasing number of ECUs that communicate with each other and external networks make the CAN bus an attractive target for cyber attackers and has led to a rise in security and privacy risks. The attack surface continues to grow proportionally to new vehicle features, and the demonstrated exploits of its vulnerabilities [1]–[6] have made vehicular security a concern for all. Securing CAN is a significant step toward ensuring that the critical systems which communicate on the bus are protected from cyber and physical attacks. Approaches toward CAN security include deploying an intrusion detection system (IDS) [7], [8] on the network to detect indications of attacks over the CAN bus, or using cryptographic techniques [9]–[11] and authentications mechanisms [12]–[14]. Although an IDS can be adopted without perturbing bus performance [15], the latter approaches cannot be easily applied to in-vehicle networks due to computational constraints and real-time requirements [16]. Hence, an IDS is a promising method for improving CAN security.

Despite their promise, adoption of IDSs poses significant challenges because their efficacy is unclear due to variations in the datasets, evaluation metrics, and experimental infrastructure used to validate proposed approaches. Also, it is often infeasible to properly vet these approaches since the documentation of their implementation and experimentation are not comprehensive enough for comparative study: reported success is often dependent on experimental setup procedures. To address these challenges, we introduce a framework to facilitate comparing CAN IDS performance in detecting attacks. Using this framework, we investigate the differences in the experimental settings of detection methods proposed in prior work and conduct a comprehensive evaluation of their performance using a consistent experimental methodology.

The contributions of this paper are:

- design and implementation of a novel framework to facilitate consistent, fair comparisons of proposed CAN IDS approaches;
- categorization and explanation of prior approaches proposed for CAN IDS;
- and application of the framework to proposed IDS approaches in eight categories.

II. Background and Related Work

Many IDSs have been proposed and developed to identify attacks on the CAN bus, but inconsistencies in their experimental evaluation hinders the adoption of these approaches in practice as their comparative performance is unclear. Further, the lack of predictable performance of these algorithms across datasets and environments, including software versions and host machine architecture, has resulted in difficulties in the interpretation of comparative study results. When an algorithm is trained and evaluated on a specific dataset and metric, its performance in other contexts remains opaque. To effectively compare proposed IDS algorithms, we have created a framework for CAN IDS evaluation that aims to eliminate the inconsistencies and enable a smooth, scalable, consistent approach for comparison of each algorithm. In the remainder of this section we discuss these inconsistencies and other drawbacks of prior work in the area of CAN IDS evaluation.

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.2022.23031
www.ndss-symposium.org

1) Disparate Training Dataset: The process used by IDSs to identify attacks is heavily influenced by the nature of the attack-free dataset used to train the underlying classification algorithms. Inconsistencies in the training datasets and assumptions used by the algorithms proposed to detect CAN bus attacks leads to significant impacts on the performance evaluation in the detection (testing) phase. For example, Marchetti and Stabili [17] used a CAN dataset containing recurring patterns within the sequence of message identifiers (IDs) or CAN IDs to train their proposed algorithm. Unfortunately, these repetitive patterns of CAN IDs are not always evident in practice, leading to errors in the algorithm's detection phase. Another study [18] on CAN bus IDS trains the system using 35 distinct datasets from varied driving behaviors. These variations in training environments have a significant impact on the detection phase.

2) Disparate Evaluation Dataset: Each proposed CAN bus IDS goes through an evaluation phase where it gets tested with a CAN dataset containing attack messages. The efficacy of the tested algorithm is highly dependent on the dataset used to evaluate it. For example, Stabili et al. [19] evaluated their proposed algorithm on a dataset containing fuzzy and replay attacks assuming that the attack message will be inserted after a predetermined number of regular messages; Marchetti and Stabili [17] used datasets with bad and mixed injection attacks; and Islam et al. [20] used denial-of-service (DoS) and spoofing attack datasets. Comparing the performance of these algorithms is difficult due to these differences in the test datasets used for their original evaluation.

3) Disparate Evaluation Metrics: The evaluation of CAN bus IDS requires the selection of appropriate metrics to demonstrate the algorithm performance. However, in most studies, there are inconsistencies among the chosen metrics. For instance, Olufowobi et al. [21] calculated true-positive rate, false-positive rate whereas Marchetti and Stabili [17] used detection rate to show the efficiency of the proposed algorithm.

Previous studies have developed frameworks for comparing anomaly detections and datasets used for IDS in CAN. Stabili et al. [22] presented a benchmark framework for CAN IDSs, which allows evaluation and comparison of four detection algorithms. Similarly, Dupont et al. [23] provided an evaluation framework to compare existing network intrusion detection mechanisms for the CAN bus. Costa [24] designed a framework for testing machine learning (ML) IDSs to determine the best algorithm that alerts the adversary when a failure in the integrity of the CAN data is identified. However, the proposed frameworks in the literature can only compare a fixed number of algorithms using the datasets provided in the framework or a particular IDS approach. We present a flexible framework that can be used to evaluate the existent state-of-the-art algorithms focusing on classifier performance with extensible plug-in capabilities for new datasets and algorithms.

III. The Framework

The evaluation of IDSs proposed for the CAN bus often depends on different performance metrics. Comparing distinct algorithms with different metrics does not provide a fair judgment of the detection accuracies as attack settings of the datasets and the objectives vary. Hence, we introduce a framework to comprehensively and rigorously analyze IDS algorithms using common input datasets and evaluation metrics. With this framework, it is possible to train all the algorithms using the same attack-free datasets, test the algorithms' attack detection mechanism using the same attack dataset, and evaluate the efficiency of the algorithms by comparing measurement values for the same metrics, thereby providing a level playing field for comparison of IDS algorithms.

The design of our framework follows the architecture presented in Figure 1. The framework transforms raw datasets into feature-rich data points in a preprocessing stage, which is described in Section IV. This stage is then followed by training the algorithms with the preprocessed data. Timing and statistical algorithms work by computing the threshold criteria for anomaly detection using the attack-free dataset, while ML algorithms utilize a part of the attack dataset—the training set—to learn patterns of the attacks. The trained ML model is then tested on another set of input from the attack set, called the testing set, to check the performance of the model. Splitting the two sets helps to avoid bias. The non-ML algorithms are tested on the entire attack dataset for anomaly detection. Finally, the evaluated results are presented using classification metrics, i.e., recall, precision, false-positive rate, accuracy, and F1 score. The remainder of this section further details the key components of the framework.

A. Data

The initial datasets used for this framework are from messages logged through the OBD-II port of a vehicle and captured in datasets provided by the Hacking and Countermeasure Research Lab [25] that contains attack-free and four message injection attacks used to compromise the vehicle's operation. These attacks are denial-of-service (DoS), fuzzy, and spoofing of IDs related to the vehicles' RPM and gear. Each dataset has a record of each message: its timestamp, CAN ID, the data length code ranging from 0 to 8, up to 8 bytes of data, and a flag field indicating attack-free or compromised message. The DoS dataset is injected with high priority messages of ID 0000 after every 0.3 milliseconds. The fuzzy dataset is injected with random IDs and data every 0.5 milliseconds, while the spoofing dataset is injected with messages every millisecond. Table I shows an overview of the total number of messages contained in each dataset.

TABLE I
Overview of the dataset.

Attack Type	Normal Messages	Injected Messages	Total Messages
DoS	3,078,250	587,521	3,665,771
Fuzzy	3,347,013	491,847	3,838,860
Gear Spoofing	3,845,890	594,252	4,443,142
RPM Spoofing	3,966,805	654,897	4,621,702

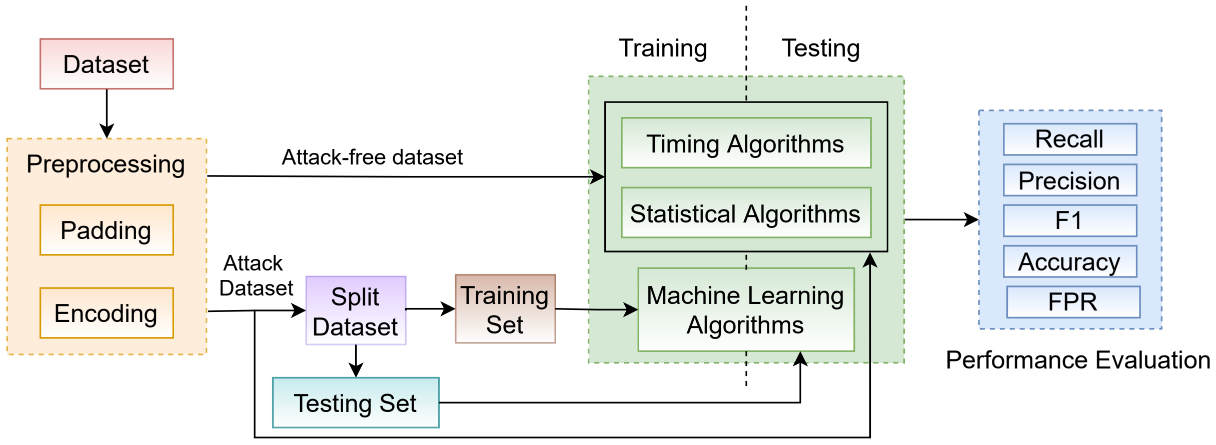

Fig. 1. Framework architecture.

B. Attack Scenarios

To evaluate the effectiveness of the algorithms considered in this paper, we consider the following attacks:

1) Denial-of-Service (DoS) Attack: The CAN bus is made unavailable to legitimate nodes that intend to send messages by flooding the bus with many messages of higher priority.

2) Fuzzy Attack: Randomly forged messages are injected into the bus aiming to cause malfunctions in ECUs.

3) Spoofing Attack: The attacking node fabricates messages normally sent by other nodes.

C. IDS Algorithms

Here, we briefly discuss the algorithms considered for this framework. We broadly classify these algorithms as timing, statistical, and ML-based.

1) Timing-based: During normal operations of a vehicle, many message IDs have a regular frequency since messages can be transmitted periodically, sporadically, or aperiodically. Periodic message instances arrive at a regular interval with a fixed length called a period. Sporadic messages recur with a minimum inter-arrival time between successive instances, while aperiodic message instances occur at arbitrary times or possibly just once during each vehicle operation, e.g., when the engine starts. However, when there is an attack, the frequencies of the IDs affected will change [7]. Detection algorithms based on timing use the inter-arrival time of messages to detect anomalies in the network. In this section, we consider two timing-based algorithms.

i. **Timing Interval:** Moore et al. [26] used inter-signal arrival time to detect anomalies in CAN signals. The algorithm evaluates the time difference, Δ_i, between the occurrences of each CAN ID using $\Delta_i = t_i - t_{i-1}$, and the average timing interval, μ, is then determined using $\mu = \sum_{i=1}^{n} \Delta_i / n$. Where n is the total number of occurrences of a particular CAN ID, t_i is the time of the i'th occurrence, and t_{i-1} is the time of the occurrence of the ID before i. The maximum observed error, e, from the expected interval is given by $e = \max_i |\Delta_i - \mu|$ and a threshold value $\alpha = e + 0.15 * \mu$ is used to detect anomalies.The threshold is determined for each ID in the attack-free dataset and then compared with the timing interval for IDs in the compromised dataset. An alarm is raised if the observed error in the compromised dataset is more than the threshold calculated from the attack-free dataset.

ii. **Frequency:** Young et al. [27] implemented an IDS by detecting changes in message frequency of particular messages over a time window t. If t is a fixed time interval for a set of m messages, the frequency, f is the rate of m transmitted in time t, i.e., $f = m/t$. The algorithm processes the attack-free dataset to determine the expected frequency for each ID. For the compromised dataset, an alarm is raised if the frequency exceeds twice the expected frequency rate from the attack-free dataset.

2) Statistical-based: Statistical-based algorithms detect anomalies based on statistical models built from the attack-free dataset. These algorithms monitor messages to discover deviations from statistical thresholds determined during normal vehicular operation. These thresholds are determined using metrics such as mean, median, mode, variance, and standard deviation. We consider five statistical-based algorithms.

i. **Cumulative Sum (CUSUM):** Olufowobi et al. [21] proposed a cumulative sum change-point algorithm for CAN that detects abrupt changes in the frequency of a message sequence. A change in message sequence is modeled using two hypotheses, θ_0 and θ_1. The hypotheses represent the statistical distribution of the message sequences before and after a change occurs. Let $D_i = \{D_1, D_2, D_3, ..., D_n\}$ be a random set of independent and uniformly distributed sequential CAN messages and has a probability density function (PDF), $pdf(D_i, \theta)$ with a known mean μ and variance σ^2. If θ is changed at time t, then $\theta = \theta_0$ before change, and $\theta = \theta_1$ after change. The instantaneous log-likelihood ratio is given by

$$S_i = (\frac{\mu D_i - \mu D_0}{\sigma_D^2})(D_i - \frac{\mu D_i + \mu D_0}{2}). \qquad (1)$$

Using the cumulative sum equation $S_n = \sum_{i=0}^{n} S_i$, real-time detection in CAN bus is found using $S_n = S_{n-1} + S_n$. An attack is detected when there is a change in the process mean after five average run length.

ii. **Entropy:** Müter and Asaj [28] proposed an intrusion detection method based on the entropy change in CAN ID. To calculate entropy, the authors used the entropy function

$$H(X) = \sum_{x \epsilon C_x} P(x) log \frac{1}{P(x)} \tag{2}$$

where $P(x)$ is the probability of the CAN ID x within a specified interval of time and C_x is the list of the CAN IDs in that timing window. After computing the entropy in the timing windows from an attack-free dataset, the IDS observes the entropy differences in the attack dataset. However, this entropy difference could only detect the presence of an attack. To specify the particular CAN ID, the authors utilize the relative entropy changes of CAN IDs among the same timing window. The equation for computing relative entropy is

$$RelEnt(p/q) = \sum_{x \epsilon C_x} p(x) log \frac{p(x)}{q(x)} \tag{3}$$

where q and p are the probability distribution of CAN IDs in the attack-free and attack scenario, respectively.

iii. **Graph Based:** Islam et al. [20] proposed a graph-based algorithm using statistical analysis that transform the CAN bus messages into a graph structure. The algorithm creates a graph for each window of 200 messages. This window size is defined for design robustness and can be changed. For each window, the common graph properties such as node (message ID) count, edge (ID sequence) count, and maximum degree (number of IDs sequential to current ID) are derived and the algorithm computes the χ^2 value, then the threshold value for the population window. The chi-squared test can be described by the following equation:

$$\chi^2_{DoF} = \sum_{i=0}^{DoF} \frac{(O_i - E_i)^2}{E_i} \tag{4}$$

where, O is the observed frequency and E is the expected frequency. DoF is the degree of freedom which can be calculated by $DoF = (i-1)(j-1)$, where i and j are the total number of message ID sequences and the sample size, respectively. The first stage of the detection phase is to compute the chi-squared value for the test dataset, denoted as "t" in $\chi^2_t > Th_{af}$, and compare it to the threshold calculated for the attack-free dataset, referred as to "af". Then the second stage is to perform the median test using $Median_t > (Median_{af} + SD_{af})$, where SD is the standard deviation. If either of these equations holds, there is an anomaly in the test dataset that gets flagged.

iv. **ID Sequences:** Marchetti and Stabili [17] proposed an algorithm that monitors the sequence of unique IDs in the attack free dataset to create a transition matrix from the recurring ID patterns. The transition matrix represents a data structure that recognize all the transitions that occurred in an attack free scenario. In a dataset with m unique IDs, the transition matrix is a square matrix of size $m \times m$, with all elements initialized to *False* (i.e., no legitimate transition is detected). For a message ID a followed by message ID b in the attack free dataset, row a and column b of the matrix is changed to *True*. The final transition matrix containing *True* and *False* values is used to detect attacks in the compromised dataset. If any of two consecutive IDs are not present in the transition matrix, an alarm is raised. Also, if the two IDs are present but there are no legitimate transitions indicated by *True* in the transition matrix, an alarm is raised.

v. **Hamming Distance:** Stabili et al. [19] used the Hamming distance, which two strings by evaluating the minimum number of substitutions that converts one of the strings to the other to detect anomalies. The Hamming distance between two strings a and b of equal length l can be evaluated using $H_l(a,b) = \sum_{i=1}^{l} |a_i - b_i|$. For CAN datasets containing a maximum of 64 bits of payload, the Hamming distance, H_l between the payloads of two consecutive instances of an ID is evaluated as

$$H_l(P_t, P_{t+1}) = \sum_{i=1}^{64} P_{i,t} \otimes P_{i,t+1}. \tag{5}$$

Here $P_{i,t}$ and $P_{i,t+1}$ are bits of the payload of an ID i at time t and $t+1$ respectively. The Hamming distance is determined by summing the results of the bitwise XOR operation between the individual bits of the payloads. An alarm is raised if the Hamming distance computed from the compromised dataset is outside the range of the calculated Hamming distances of the same ID from the attack-free dataset.

3) ML-Based: IDSs based on ML identify patterns in CAN transmissions to classify messages and flag intrusions with little human intervention [29]. ML-based algorithms can be unsupervised or supervised. A supervised algorithm requires labeled data for training, while an unsupervised algorithm finds patterns in unlabeled data. We consider artificial neural network (ANN), a machine learning algorithm modeled to imitate biological neurons in our framework.

Artificial Neural Network: Paul and Islam [30] proposed an anomaly detection based on ANN architecture consisting of 4 layers—an input layer, two hidden layers, and an output layer. The input layer receives a preprocessed dataset containing ID fields and 8 bytes of data fields. Each hidden layer consists of 16 neurons that evaluate the summation of the multiplication of weights and inputs. The result from each layer passes to an activation function that determines the values fed as inputs to the next layer. For anomaly detection, the output of ANN gives a binary classification of either 1 or 0 to signify legitimate messages and compromised messages, respectively.

IV. Experimental Validation

To evaluate the performance of the algorithms in our benchmark framework, we used real vehicle CAN traffic,

TABLE II
COMPARATIVE ANALYSIS OF MENTIONED ALGORITHMS BASED ON PROPOSED DECOUPLED FRAMEWORK

Metrics	Time Interval				Frequency				CUSUM				Entropy				Graph				ID Sequences				Hamming				ANN			
	[7], [31], [32]				[27], [33]				[21]				[18], [28], [34]				[20], [35]				[17], [36]				[19]				[30]			
Attack Type	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
Recall	0.83	0.97	0.0	0.0	0.18	0.99	0.05	0.07	1.0	0.99	1.0	1.0	0.02	0.03	0.06	0.06	0.69	0.98	0.75	0.53	0.96	1.0	0.33	0.42	0.95	0.99	0.0	0.0	1.0	0.98	0.86	0.87
Precision	0.60	0.51	0.0	0.0	0.99	1.0	0.89	0.9	1.0	0.48	1.0	1.0	0.25	0.33	016	0.21	0.98	0.99	0.99	0.99	0.57	0.63	0.41	0.59	0.10	0.99	0.0	0.0	0.99	1.0	1.0	1.0
F1	0.70	0.67	0.0	0.0	0.3	0.99	0.1	0.12	1.0	0.65	1.0	1.0	0.03	0.05	0.08	0.1	0.81	0.99	0.86	0.69	0.71	0.78	0.37	0.12	0.97	0.99	0.0	0.0	0.99	0.99	0.92	0.93
Accuracy	0.95	0.96	0.83	0.83	0.86	1.0	0.86	0.86	1.0	0.64	1.0	1.0	0.63	0.68	0.74	0.71	0.80	0.99	0.82	0.67	0.87	0.92	0.83	0.87	0.99	0.99	0.85	0.86	0.99	0.98	0.86	0.87
FPR	0.03	0.04	0.02	0.02	0.0	0.0	0.0	0.0	0.0	0.45	0.0	0.0	0.03	0.02	0.08	0.08	0.02	0.0	0.01	0.02	0.15	0.09	0.08	0.0	0.0	0.0	0.0	0.0	0.06	0.0	0.0	0.0

* 1 = DoS Attack, 2 = Fuzzy Attack, 3 = RPM Spoofing Attack, 4 = Gear Spoofing Attack

which contains DoS attacks, fuzzy attacks, spoofing attack datasets. These datasets are preprocessed for use by all the algorithms in the framework. In these datasets, the flag "R" represents an attack-free message, while "T" represents a compromised message. We trained the statistical and timing detection methods using the attack-free dataset, while the machine learning algorithm was trained using 80% of the compromised datasets. Furthermore, we have retained most of the original settings for the parameters in the ANN model and introduced batch normalization and dropout layers after the hidden layers to reduce overfitting.

To evaluate the algorithms, we used standard metrics, such as recall, precision, F1 score, accuracy, and false-positive rates (FPR). The recall is the proportion of correct attack predictions in the attack class, given by $R = \frac{TP}{(TP+FN)}$, where TP and FP are the true positive and false negative values respectively. The precision gives the proportion of identified attacks that were actually correct, denoted as $P = \frac{TP}{(TP+FP)}$ where FP is the false negative value. F1 score gives a view of the weighted average of the precision and recall, given by: $F1 = \frac{2*(Recall*Precision)}{Recall+Precision}$. Accuracy, $A = \frac{TP+TN}{TP+FP+TN+FN}$, is the proportion of right predictions in the total observations and TN is the true negative value. FPR represent the probability of raising a false alarm for an attack, calculated as $FPR = \frac{FP}{(FP+TN)}$.

V. DISCUSSION

Table II shows the performance of different approaches on the dataset. We observe that each method has its merits and weaknesses, and performance varies by the basis of the attack. On fuzzy attacks, the timing interval algorithm works best as detection of an anomaly becomes easier when there is an aberration from the expected pattern of a CAN ID. However, sole reliance on CAN IDs makes the performance of the same approach questionable in detecting an attack of different forms, such as spoofing. Straightforward techniques, such as measuring Hamming distance and calculating entropy changes, are computationally inexpensive and work well on DoS attack detection where there is no reliance on different attributes of an ECU. However, these algorithms lack the required sophistication for detecting interrelated relationships in vehicular features, i.e., dependencies between CAN IDs that cause them to change transmission behavior with respect to each other and the physical state of the vehicle. CUSUM provides a frequency-based statistical approach that gives a consistent performance in detecting attacks. However, more generalized approaches can detect additional forms of attacks. Our understanding suggests that graph-based algorithms [20] and ML approaches such as ANN [30] can discover better correlations among the features, which in turn helps in improving anomaly detection. ML approaches can also benefit from automation of parameter tuning, i.e., feature selection, finding optimum threshold values for detection, average intervals, or window sizes [37]. Generally the ML models have better performance with more computational resources and feature-rich datasets, but these may not be easily available.

In light of our observations, we suggest the methods that consider the interrelationship between messages, such as ML models and graph-based models, are better suited for the CAN bus datasets. In addition to comparing the performance of algorithms for IDS in CAN, our framework provides a standard evaluation platform that allows seamless incorporation of new algorithms that can be tested against the algorithms already in the framework. With a preprocessing stage, the framework also allows the use of new datasets to evaluate the performance of the algorithms already implemented.

VI. CONCLUSION

This paper presents a framework for benchmarking CAN IDS algorithms to compare them using consistent metrics, which facilitates fair and reproducible experiments. Future work can add more IDSs [38], [39], attacks [40], datasets [41]–[43], metrics [44], and runtime performance evaluation.

ACKNOWLEDGMENT

This work is supported in part by NSF grants OAC-2001789, CNS-2046705, and Colorado State Bill 18-086.

REFERENCES

[1] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle." BlackHat USA, 2015.
[2] S. Hounsinou, M. Stidd, U. Ezeobi, H. Olufowobi, M. Nasri, and G. Bloom, "Vulnerability of controller area network to schedule-based attacks," in *2021 IEEE Real-Time Systems Symposium (RTSS)*. IEEE, 2021, pp. 495–507.
[3] P. Bajpai, R. Enbody, and B. H. Cheng, "Ransomware targeting automobiles," in *Proceedings of the Second ACM Workshop on Automotive and Aerial Vehicle Security*, 2020, pp. 23–29.
[4] K.-T. Cho and K. G. Shin, "Error handling of in-vehicle networks makes them vulnerable," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 1044–1055.

[5] S. Mukherjee, H. Shirazi, I. Ray, J. Daily, and R. Gamble, "Practical dos attacks on embedded networks in commercial vehicles," in *International Conference on Information Systems Security*. Springer, 2016, pp. 23–42.

[6] G. Bloom, "WeepingCAN: A Stealthy CAN Bus-off Attack," in *Workshop on Automotive and Autonomous Vehicle Security*. Internet Society, Feb. 2021.

[7] C. Young, J. Zambreno, H. Olufowobi, and G. Bloom, "Survey of automotive controller area network intrusion detection systems," *IEEE Design & Test*, vol. 36, no. 6, pp. 48–55, 2019.

[8] H. Shirazi, I. Ray, and C. Anderson, "Using machine learning to detect anomalies in embedded networks in heavy vehicles," in *12th International Symposium on Foundations and Practice of Security*, vol. 12056, 2019.

[9] J. A. Bruton, "Securing can bus communication: An analysis of cryptographic approaches," *Nat. Univ. Ireland, Galway*, pp. 1–5, 2014.

[10] G. Bella, P. Biondi, G. Costantino, and I. Matteucci, "Toucan: A protocol to secure controller area network," in *Proceedings of the ACM Workshop on Automotive Cybersecurity*, 2019, pp. 3–8.

[11] J. B. Asante, T. Rahman, T. I. Reuter, and V. Moore, "Implementing cryptographic hash functions on can bus."

[12] B. Groza and P.-S. Murvay, "Security solutions for the controller area network: Bringing authentication to in-vehicle networks," *IEEE Vehicular Technology Magazine*, vol. 13, no. 1, pp. 40–47, 2018.

[13] R. Kurachi, Y. Matsubara, H. Takada, N. Adachi, Y. Miyashita, and S. Horihata, "Cacan-centralized authentication system in can (controller area network)," in *14th Int. Conf. on Embedded Security in Cars (ESCAR 2014)*, 2014.

[14] J. Daily, D. Nnaji, and B. Ettlinger, "Securing CAN Traffic on J1939 Networks," in *Workshop on Automotive and Autonomous Vehicle Security*. Internet Society, Feb. 2021.

[15] H. Olufowobi, S. Hounsinou, and G. Bloom, "Controller area network intrusion prevention system leveraging fault recovery," in *Proceedings of the ACM Workshop on Cyber-Physical Systems Security & Privacy*, 2019, pp. 63–73.

[16] R. Kurachi, H. Takada, H. Ueda, and S. Takimoto, "Towards Minimizing MAC Utilization for Controller Area Network," in *Proceedings of the Second ACM Workshop on Automotive and Aerial Vehicle Security*, ser. AutoSec '20. New York, NY, USA: ACM, Mar. 2020, pp. 45–50.

[17] M. Marchetti and D. Stabili, "Anomaly detection of can bus messages through analysis of id sequences," in *2017 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2017, pp. 1577–1583.

[18] Q. Wang, Z. Lu, and G. Qu, "An entropy analysis based intrusion detection system for controller area network in vehicles," in *2018 31st IEEE International System-on-Chip Conference (SOCC)*. IEEE, 2018, pp. 90–95.

[19] D. Stabili, M. Marchetti, and M. Colajanni, "Detecting attacks to internal vehicle networks through hamming distance," in *2017 AEIT International Annual Conference*. IEEE, 2017, pp. 1–6.

[20] R. Islam, R. U. D. Refat, S. M. Yerram, and H. Malik, "Graph-based intrusion detection system for controller area networks," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[21] H. Olufowobi, U. Ezeobi, E. Muhati, G. Robinson, C. Young, J. Zambreno, and G. Bloom, "Anomaly detection approach using adaptive cumulative sum algorithm for controller area network," in *Proceedings of the ACM Workshop on Automotive Cybersecurity*, 2019, pp. 25–30.

[22] D. Stabili, F. Pollicino, and A. Rota, "A benchmark framework for can ids," in *Italian Conference on Cyber Security*, 2021.

[23] G. Dupont, J. Den Hartog, S. Etalle, and A. Lekidis, "Evaluation framework for network intrusion detection systems for in-vehicle can," in *2019 IEEE International Conference on Connected Vehicles and Expo (ICCVE)*. IEEE, 2019, pp. 1–6.

[24] T. Costa Cañones, "Framework for benchmarking the ids of the controller area network," Master's thesis, Universitat Politècnica de Catalunya, 2021.

[25] H. Lee, S. H. Jeong, and H. K. Kim, "OTIDS: A Novel Intrusion Detection System for In-vehicle Network by Using Remote Frame," in *2017 15th Annual Conference on Privacy, Security and Trust (PST)*, vol. 00, Aug. 2017, pp. 57–5709.

[26] 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.

[27] C. Young, H. Olufowobi, G. Bloom, and J. Zambreno, "Automotive intrusion detection based on constant can message frequencies across vehicle driving modes," in *Proceedings of the ACM Workshop on Automotive Cybersecurity*, 2019, pp. 9–14.

[28] M. Müter and N. Asaj, "Entropy-based anomaly detection for in-vehicle networks," in *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2011, pp. 1110–1115.

[29] K. Pawelec, R. A. Bridges, and F. L. Combs, "Towards a can ids based on a neural network data field predictor," in *Proceedings of the ACM Workshop on Automotive Cybersecurity*, ser. AutoSec '19. New York, NY, USA: ACM, 2019, p. 31–34.

[30] A. Paul and M. R. Islam, "An artificial neural network based anomaly detection method in can bus messages in vehicles," in *2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI)*. IEEE, 2021, pp. 1–5.

[31] M. Gmiden, M. H. Gmiden, and H. Trabelsi, "An intrusion detection method for securing in-vehicle can bus," in *2016 17th International Conference on Sciences and Techniques of Automatic Control and Computer Engineering (STA)*. IEEE, 2016, pp. 176–180.

[32] 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 *2016 international conference on information networking (ICOIN)*. IEEE, 2016, pp. 63–68.

[33] A. Taylor, N. Japkowicz, and S. Leblanc, "Frequency-based anomaly detection for the automotive can bus," in *2015 World Congress on Industrial Control Systems Security (WCICSS)*. IEEE, 2015, pp. 45–49.

[34] W. Wu, Y. Huang, R. Kurachi, G. Zeng, G. Xie, R. Li, and K. Li, "Sliding window optimized information entropy analysis method for intrusion detection on in-vehicle networks," *IEEE Access*, vol. 6, pp. 45 233–45 245, 2018.

[35] Z. Tyree, R. A. Bridges, F. L. Combs, and M. R. Moore, "Exploiting the shape of can data for in-vehicle intrusion detection," in *2018 IEEE 88th Vehicular Technology Conference (VTC-Fall)*. IEEE, 2018, pp. 1–5.

[36] A. K. Desta, S. Ohira, I. Arai, and K. Fujikawa, "Id sequence analysis for intrusion detection in the can bus using long short term memory networks," in *2020 IEEE International Conference on Pervasive Computing and Communications Workshops*. IEEE, 2020, pp. 1–6.

[37] H. Lawrence, U. Ezeobi, G. Bloom, and Y. Zhuang, "Shining New Light on Useful Features for Network Intrusion Detection Algorithms," in *2022 IEEE 19th Annual Consumer Communications Networking Conference (CCNC)*, Jan. 2022, pp. 369–377, iSSN: 2331-9860.

[38] H. Olufowobi, G. Bloom, C. Young, and J. Zambreno, "Work-in-Progress: Real-Time Modeling for Intrusion Detection in Automotive Controller Area Network," in *2018 IEEE Real-Time Systems Symposium (RTSS)*, Dec. 2018, pp. 161–164.

[39] T. Koyama, T. Shibahara, K. Hasegawa, Y. Okano, M. Tanaka, and Y. Oshima, "Anomaly detection for mixed transmission can messages using quantized intervals and absolute difference of payloads," in *Proceedings of the ACM Workshop on Automotive Cybersecurity*, ser. AutoSec '19. New York, NY, USA: ACM, 2019, p. 19–24.

[40] H. Olufowobi and G. Bloom, "Chapter 16 - Connected Cars: Automotive Cybersecurity and Privacy for Smart Cities," in *Smart Cities Cybersecurity and Privacy*, D. B. Rawat and K. Z. Ghafoor, Eds. Elsevier, Jan. 2019, pp. 227–240.

[41] M. E. Verma, M. D. Iannacone, R. A. Bridges, S. C. Hollifield, B. Kay, and F. L. Combs, "Road: the real ornl automotive dynamometer controller area network intrusion detection dataset (with a comprehensive can ids dataset survey & guide)," *arXiv:2012.14600*, 2020.

[42] D. Blevins, P. Moriano, R. A. Bridges, M. E. Verma, M. D. Iannacone, and S. C. Hollifield, "Time-based can intrusion detection benchmark," in *Workshop on Automotive and Autonomous Vehicle Security*. Internet Society, Feb. 2021.

[43] U. Ezeobi, H. Olufowobi, C. Young, J. Zambreno, and G. Bloom, "Reverse Engineering Controller Area Network Messages using Unsupervised Machine Learning," *IEEE Consumer Electronics Magazine*, pp. 1–1, 2020, conference Name: IEEE Consumer Electronics Magazine.

[44] 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.