CANLP: NLP-Based Intrusion Detection System for CAN

Kavya Balasubramanian
University of Washington
Seattle, USA
kavyab25@uw.edu

Adithya Gowda Baragur
University of Washington
Seattle, USA
gowdabar@uw.edu

Denis Donadel
University of Padua
Padua, Italy
donadel@phd.unipd.it

Dinuka Sahabandu
University of Washington
Seattle, USA
sdinuka@uw.edu

Alessandro Brighente
University of Padua
Padua, Italy
alessandro.brighente@unipd.it

Bhaskar Ramasubramanian
Western Washington University
Bellingham, USA
ramasub@wwu.edu

Mauro Conti
University of Padua
Padua, Italy
Delft University of Technology
Delft, Netherlands
mauro.conti@unipd.it

Radha Poovendran
University of Washington
Seattle, USA
rp3@uw.edu

ABSTRACT

The Controller Area Network (CAN) protocol is the most widely used standard for in-vehicle networks. However, the CAN protocol lacks essential security features (e.g., encryption) which makes it vulnerable to exploits by an adversary. In this paper, we develop CANLP, a Natural Language Processing (NLP)-based intrusion detection system to find whether a transmitted message originated from a legitimate ECU or an adversary. CANLP uses the Term Frequency-Inverse Document Frequency (TF-IDF) to discern complex features associated with CAN data and trains ML models with these features to identify fuzzing, spoofing, and masquerade attacks. When an attack is detected, CANLP identifies the specific ECU on which the attack was mounted on. Through extensive experiments on four public vehicle network datasets, we show that CANLP performs attack classification with high F1-score of 0.9974. We also demonstrate using a testbed that CANLP can be deployed for attack detection on resource-constrained hardware a latency of < 0.05 *ms*.

KEYWORDS

Controller Area Network, Natural Language Processing, TF-IDF

ACM Reference Format:
Kavya Balasubramanian, Adithya Gowda Baragur, Denis Donadel, Dinuka Sahabandu, Alessandro Brighente, Bhaskar Ramasubramanian, Mauro Conti, and Radha Poovendran. 2024. CANLP: NLP-Based Intrusion Detection System for CAN. In *Proceedings of ACM/ SIGAPP Symposium on Applied Computing, Avila, Spain, April 8-12, 2024 (SAC'24),* 3 pages.
https://doi.org/10.1145/3605098.3636083

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
SAC'24, April 8-12, 2024, Avila, Spain
© 2024 Association for Computing Machinery.
ACM ISBN 979-8-4007-0243-3/24/04…$15.00
https://doi.org/10.1145/3605098.3636083

1 INTRODUCTION

The Controller Area Network (CAN) protocol for in-vehicle messaging standards lacks critical security features and can be exploited to compromise Electronic Control Units (ECUs), potentially affecting driver and passenger safety. Intrusion Detection Systems (IDSs), particularly based on Machine Learning (ML), offer a scalable solution to these security challenges.

However, deploying ML-bassed IDSs in real-time can be challenging since: (i) Stealthy attacks mimic the frequency of legitimate CAN messages, making them undetectable by timing-based IDSs; (ii) IDSs analyzing a window of messages are unable to identify the specific ECU and frame of attack; (iii) CAN transmission speeds (up to 1 Mbps) require low latency attack prediction which hinder the deployment of large ML-based IDSs using limited hardware.

In this paper, we develop CANLP, a NLP-Based Intrusion Detection System for CAN, which uses the TF-IDF technique [1] to group binary CAN data into bits of different lengths and extract fine-grained features for real-time attack detection. We exploit the structure of CAN messages to construct a human-readable format according to the CAN Database (DBC) file and interpret this using features obtained via Natural Language Processing (NLP) based techniques. CANLP uses the features to train a Deep Neural Network (DNN) to distinguish between legitimate transmissions and adversarial behavior to perform *multi-class attack classification*. Our feature extraction demonstrates variability in features among different attacks, thus allowing us to use a small DNN model, which is further reduced in size using quantization [3]. This enables CANLP to perform attack detection in real-time with low overhead.

We evaluate the effectiveness of CANLP through extensive experiments on four publicly available vehicle network datasets. Our IDS deployment on a CAN testbed show that CANLP detects fuzzing, spoofing, or masquerade attacks with a high F1-score of 0.9974 even after model compression. We implement CANLP on a Raspberry Pi 4 Model B running at 1.8GHz with 1GB of RAM achieving classification latency as low as 0.049ms.

2 THREAT AND DEFENSE MODELS

Table 1: Types of attacks that an adversary can carry out on CAN Bus and importance of attributes (ID, Data, and Timestamp) of malicious CAN packets for attack identification. Symbols ●, ◐, and ○ represent full, partial, or no dependency on the particular attribute. CANLP detects Timing Opaque attacks using feature extraction techniques on ID and Data.

Attack Type	Attack	ID	Data	Timestamp
Timing Transparent	Replay	◐	◐	●
	DoS	◐	○	●
Timing Opaque	Spoofing	●	◐	○
	Fuzzing	●	●	○
	Masquerade	●	◐	○

2.1 Adversary Assumptions

We assume that the adversary can intercept all CAN traffic and has ability to inject messages of their own into the bus. The adversary performs attacks through physical access by introducing a new ECU to the bus through the vehicle's OBD-II port or taking control of a compromised ECU [2]. We consider different levels of compromised ECUs based on the control that the adversary exerts [13].

2.2 Attack Scenarios

The objective of the adversary is to compromise the safety of the vehicle by injecting anomalous messages into the CAN bus by carrying out different classes of CAN attacks shown in Table 1. Attacks can broadly be classified into *timing transparent (TT)* and *timing opaque (TO)* attacks. TT (TO) attacks are detectable (not detectable) using a frequency-based method [13].

Due to the nature of TO attacks, specific methods such as a payload-based detector that uses data field, or a side-channel method monitoring the physical layer must be deployed [13]. In this paper, we design CANLP to detect fuzzing, spoofing, and masquerade TO attacks using feature extraction techniques on the CAN payload. The adversary is assumed to perform multiple attacks in a sporadic manner with no observable pattern in attack frequency, while targeting any component of the vehicle to cause abnormal behavior.

2.3 Defense Model

The defense has limited real-time computational resources and uses ML to train a model or re-train publicly available models to achieve high F1-score for attack classification. We assume that the defense is aware that the adversary performs different types of attacks on the CAN Bus and has knowledge of these attacks, but has no knowledge of the order or frequency in which these attacks are performed. The defense has access to a set of CAN packets whose labels are known for model training. Our goal is to develop an IDS that detects fuzzing, spoofing and masquerade attacks on the CAN Bus in real-time and can be deployed inside a vehicle.

3 DATASETS AND CANLP IMPLEMENTATION

We use four publicly available CAN Datasets to evaluate CANLP: (i) HCRL Car Hacking: Attack and Defense Challenge (AD) [7], (ii) HCRL Car-Hacking (CH) [11], (iii) HCRL Survival Analysis Dataset for Automobile IDS (SA) [4], and (iv) Real ORNL (Oak Ridge National Laboratory) Automotive Dynamometer (ROAD) CAN Intrusion Dataset [13]. We carry out necessary preprocessing to remove duplicate CAN frames and to balance the dataset in terms of number of samples considered from each class.

Table 2: Number of features and number of trainable DNN parameters for 1-gram, (1,2)-gram and (1,2,3)-gram Character (Char) level TF-IDF features.

Granularity of Features	No of TF-IDF Features	Trainable Parameters
Char 1-gram	16	10755
Char (1,2)-gram	272	43523
Char (1,2,3)-gram	4368	567811

A raw CAN data frame is transmitted in a hexadecimal format which can be decoded using the CAN DBC in a bitwise manner to interpret signal definition as physical parameters [6]. The data in a CAN frame can be segmented into blocks of 8 one-byte values, 64 one-bit values, 1 sixty-four bits value, or any combination of these [6]. Therefore, we choose a character as a term while using N-gram TF-IDF for extracting features. Our proposed character-level CAN data features include all (1,2)-gram byte patterns.

Adversary-manipulated CAN data related to TO attacks might exhibit different bit orderings compared to the patterns found in normal CAN messages. We observe that such consecutive bit patterns of length N can be captured through N-gram units [10]. The total number of required TF-IDF features to represent each CAN frame and number of trainable DNN parameters are shown in Table 2.

The features extracted from CAN frames are used to train five ML models for classification, including Support Vector Machine (SVM), Logistic Regression (LR), Gaussian Naive Bayes (GNB), Decision Tree (DT), Random Forest (RF) [5]. We also train a DNN to effectively extract features at different levels of abstraction. The DNN learns complex patterns compared to traditional ML algorithms [9] allowing easy deployment on resource-constrained hardware [12].

4 EXPERIMENTS AND TESTBED EVALUATION

We evaluated CANLP on four datasets using five different ML algorithms- SVM, LR, GNB, DT, and RF - and the DNN and computed the F1-score. Our results are shown in Table 3. Since DNN achieves best performance compared to other models (highest F1-score for three datasets and close to best F1-score for the other), we choose DNNs for deploying CANLP on the testbed.

To prove feasibility of deploying CANLP in a real scenario, we implemented it on a CAN testbed supporting bandwidths upto 1Mbps, shown in Fig. 1. To simulate bus traffic from different ECUs, we used a Raspberry Pi 3 Model B equipped with a PiCAN2 Duo hat and program it to replay one dataset at a time. A comprehensive analysis of CANLP and for other SOTA hardware-deployed models is presented in Table 4 when tested for F1-score and latency. Our processing delay is within the acceptable processing delays of <10 *ms* [15] for a standard CAN bus with a baud rate of upto 1Mbps with both the Raspberry Pi 3 and Raspberry Pi 4. The results of our study emphasize that CANLP demonstrates superior performance compared to the SOTA models in latency, while maintaining a comparable F1-score across all attack scenarios.

Table 3: F1-scores (F1) for four different datasets for five machine learning models -Support Vector Machines (SVM), Logistic Regression (LR), Gaussian Naive Bayes (GNB), Decision Trees (DT), Random Forest (RF), and a simple Deep Neural Network (DNN). We observe that the DNN showcases the best performance compared to other models, exhibiting higher average F1-score for three out of five datasets and close to best F1-score for the remaining datasets.

Model Type	Model	F1 AD (Driving)	F1 AD (Stationary)	F1 CH	F1 SA	F1 ROAD
Machine Learning	SVM	0.815	0.921	0.971	0.873	0.982
	LR	0.816	0.925	0.971	0.884	0.976
	GNB	0.737	0.836	0.865	0.787	0.934
	DT	0.967	0.961	**0.998**	0.974	0.991
	RF	**0.968**	0.979	0.995	0.973	0.995
Deep Learning	**DNN**	0.964	**0.979**	0.997	**0.976**	**0.997**

Table 4: This table shows the F1-Score and mean and standard deviation (Std) of latency for CANLP and other SOTA hardware-deployed models when tested on the CH dataset. Here, NR refers to Not Reported. The F1-score and latency of the models which perform binary classification (MTH-IDS and MA-QCNN) have been averaged and summed up respectively across all the attacks for fair comparison against the multi-class classification models presented in ACGAN and CANLP (Our Work). While the F1-scores cannot be directly compared due to the incorporation of synthesized data into our dataset, the achieved F1-score remains comparable to those achieved by other SOTA models. Our findings highlight that CANLP outperforms SOTA models in terms of latency and maintains a comparable F1-score across all attacks. Consequently, these results underscore CANLP as the optimal choice for real-time deployment.

Method	Hardware Platform	CPU	F1-Score	Latency	
				Mean	Std
MTH-IDS [14]	Raspberry Pi 3 Model B	1.2 GHz Broadcom BCM2837	0.999	1.722 ms	NR
MA-QCNN [8]	Zynq FPGA	1.3 GHz Arm Cortex-A53 MPCore	0.996	1.290 ms	NR
ACGAN [15]	Raspberry Pi 4 Model B	1.8 GHz ARM Cortex-A72	0.992	0.538 ms	0.030 ms
CANLP	**Raspberry Pi 3 Model B**	**1.2 GHz Broadcom BCM2837**	**0.997**	**0.213 ms**	**0.030 ms**
CANLP	**Raspberry Pi 4 Model B**	**1.8 GHz ARM Cortex-A72**	**0.997**	**0.049 ms**	**0.011 ms**

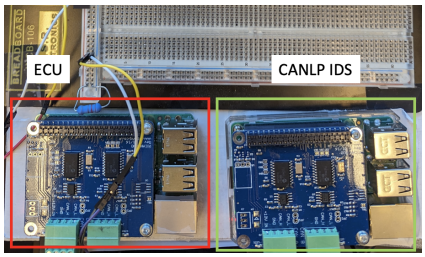

Figure 1: A picture of our CAN bus testbed. It has a Raspberry Pi 3 Model B equipped with a PiCAN2 Duo hat (Red) to send real-world traffic to the bus. A Raspberry Pi 4 Model B with PiCAN2 Duo hat (Green) reads the data sent through CAN Bus and uses CANLP to make the attack predictions.

ACKNOWLEDGMENTS

This work was supported by the US National Science Foundation, the Office of Naval Research, European Commission under the Horizon Europe Program, as part of the project LAZARUS, via grants CNS-2153136, N00014-20-1-2636, N00014-23-1-2386, and 101070303 respectively. We thank Omitech S.r.l. for supporting the research of Denis Donadel. The content of this article does not reflect the official opinion of the European Union. Responsibility for the information and views expressed therein lies entirely with the authors.

REFERENCES

[1] Akiko Aizawa. 2003. An information-theoretic perspective of tf–idf measures. *Information Processing & Management* 39, 1 (2003), 45–65.
[2] Mehmet Bozdal, Mohammad Samie, and Ian Jennions. 2018. A survey on can bus protocol: Attacks, challenges, and potential solutions. In *International Conference on Computing, Electronics & Communications Engineering*. IEEE, 201–205.
[3] Amir Gholami, Sehoon Kim, Zhen Dong, Zhewei Yao, Michael W Mahoney, and Kurt Keutzer. 2022. A survey of quantization methods for efficient neural network inference. In *Low-Power Computer Vision*. Chapman and Hall/CRC, 291–326.
[4] Mee Lan Han, Byung Il Kwak, and Huy Kang Kim. 2018. Anomaly intrusion detection method for vehicular networks based on survival analysis. *Vehicular communications* 14 (2018), 52–63.
[5] Peter Harrington. 2012. *Machine Learning in Action*. Simon and Schuster, New York, NY, USA.
[6] Influx Technology. 2021. Understanding CAN DBC. www.influxtechnology.com/post/understanding-can-dbc. Accessed: Aug. 23, 2023.
[7] Hyunjae Kang, Byung Il Kwak, Young Hun Lee, Haneol Lee, Hwejae Lee, and Huy Kang Kim. 2021. Car hacking and defense competition on in-vehicle network. In *Workshop on Automotive and Autonomous Vehicle Security*, Vol. 2021. 25.
[8] Shashwat Khandelwal and Shanker Shreejith. 2022. A Lightweight Multi-Attack CAN Intrusion Detection System on Hybrid FPGAs. In *2022 32nd International Conference on Field-Programmable Logic and Applications (FPL)*. 425–429.
[9] Xiao-Yang Liu, Yiming Fang, Liuqing Yang, Zechu Li, and Anwar Walid. 2022. Chapter 9 - High-performance tensor decompositions for compressing and accelerating deep neural networks. In *Tensors for Data Processing*, Yipeng Liu (Ed.). Academic Press, 293–340. https://doi.org/10.1016/B978-0-12-824447-0.00015-7
[10] Dinuka Sahabandu, Sukarno Mertoguno, and Radha Poovendran. 2023. A Natural Language Processing Approach for Instruction Set Architecture Identification. *IEEE Transactions on Information Forensics and Security* (2023).
[11] Eunbi Seo, Hyun Min Song, and Huy Kang Kim. 2018. GIDS: GAN based intrusion detection system for in-vehicle network. In *2018 16th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 1–6.
[12] Tensorflow. 2023. Tensorflow Lite. https://www.tensorflow.org/lite. Accessed: Aug. 30, 2023.
[13] Miki E Verma, Michael D Iannacone, Robert A Bridges, Samuel C Hollifield, Pablo Moriano, Bill Kay, and Frank L Combs. 2020. Addressing lack of comparability & testing in CAN intrusion detection research: A comprehensive guide to CAN IDS data & introduction of the ROAD dataset. *arXiv preprint arXiv:2012.14600* (2020).
[14] Li Yang, Abdallah Moubayed, and Abdallah Shami. 2022. MTH-IDS: A Multitiered Hybrid Intrusion Detection System for Internet of Vehicles. *IEEE Internet of Things Journal* 9, 1 (jan 2022), 616–632. https://doi.org/10.1109/jiot.2021.3084796
[15] Qingling Zhao, Mingqiang Chen, Zonghua Gu, Siyu Luan, Haibo Zeng, and Samarjit Chakrabory. 2022. CAN Bus Intrusion Detection Based on Auxiliary Classifier GAN and Out-of-Distribution Detection. *ACM Trans. Embed. Comput. Syst.* 21, 4, Article 45 (sep 2022), 30 pages. https://doi.org/10.1145/3540198