TU Delft

Delft University of Technology

CANEDERLI

On the Impact of Adversarial Training and Transferability on CAN Intrusion Detection Systems

Marchiori, Francesco; Conti, Mauro

DOI
10.1145/3649403.3656486

Publication date
2024

Document Version
Final published version

Published in
WiseML 2024 - Proceedings of the 2024 ACM Workshop on Wireless Security and Machine Learning

Citation (APA)
Marchiori, F., & Conti, M. (2024). CANEDERLI: On the Impact of Adversarial Training and Transferability on CAN Intrusion Detection Systems. In *WiseML 2024 - Proceedings of the 2024 ACM Workshop on Wireless Security and Machine Learning* (pp. 8-13). (WiseML 2024 - Proceedings of the 2024 ACM Workshop on Wireless Security and Machine Learning). Association for Computing Machinery (ACM). https://doi.org/10.1145/3649403.3656486

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.

CANEDERLI: On The Impact of Adversarial Training and Transferability on CAN Intrusion Detection Systems

Francesco Marchiori
University of Padova
Padua, Italy
francesco.marchiori.4@phd.unipd.it

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

ABSTRACT

The growing integration of vehicles with external networks has led to a surge in attacks targeting their Controller Area Network (CAN) internal bus. As a countermeasure, various Intrusion Detection Systems (IDSs) have been suggested in the literature to prevent and mitigate these threats. With the increasing volume of data facilitated by the integration of Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) communication networks, most of these systems rely on data-driven approaches such as Machine Learning (ML) and Deep Learning (DL) models. However, these systems are susceptible to adversarial evasion attacks. While many researchers have explored this vulnerability, their studies often involve unrealistic assumptions, lack consideration for a realistic threat model, and fail to provide effective solutions.

In this paper, we present **CANEDERLI** (**CAN E**vasion **D**etection **R**esi**LI**ence), a novel framework for securing CAN-based IDSs. Our system considers a realistic threat model and addresses the impact of adversarial attacks on DL-based detection systems. Our findings highlight strong transferability properties among diverse attack methodologies by considering multiple state-of-the-art attacks and model architectures. We analyze the impact of adversarial training in addressing this threat and propose an adaptive online adversarial training technique outclassing traditional fine-tuning methodologies with F1 scores up to 0.941. By making our framework publicly available, we aid practitioners and researchers in assessing the resilience of IDSs to a varied adversarial landscape.

CCS CONCEPTS

• **Security and privacy** → **Intrusion detection systems**; • **Computing methodologies** → **Machine learning**.

KEYWORDS

Controller Area Network; Intrusion Detection Systems; Adversarial Attacks; Adversarial Transferability; Adversarial Training

ACM Reference Format:
Francesco Marchiori and Mauro Conti. 2024. CANEDERLI: On The Impact of Adversarial Training and Transferability on CAN Intrusion Detection Systems. In *Proceedings of the 2024 ACM Workshop on Wireless Security and Machine Learning (WiseML '24), May 31, 2024, Seoul, Republic of Korea.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3649403.3656486

This work is licensed under a Creative Commons Attribution-NonCommercial International 4.0 License.

WiseML '24, May 31, 2024, Seoul, Republic of Korea
© 2024 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0602-8/24/05.
https://doi.org/10.1145/3649403.3656486

1 INTRODUCTION

The proliferation of advanced functionalities in modern vehicles necessitates an increased number of Electronic Control Units (ECUs). As such, communication between these components becomes vital for ensuring the reliable operation of the vehicle's systems and features. This heightened communication underscores the critical role of the Controller Area Network (CAN) bus in facilitating seamless interaction among ECUs. Furthermore, the scope of communication extends beyond the confines of the vehicle itself, including interactions with external entities such as other vehicles (V2V) and infrastructures (V2I). These communication protocols enable various functionalities, including cooperative driving, real-time traffic management, and advanced driver assistance systems [8].

This heightened connectivity also increases potential security threats, prompting the need for robust Intrusion Detection Systems (IDSs). These security tools are designed to monitor network or system activities for malicious activities or policy violations [20]. They analyze incoming network traffic, system logs, or other data sources and alert administrators or take action when they detect suspicious behavior or known attack patterns. Due to the benefits offered by data-driven methodologies, Machine Learning (ML) and Deep Learning (DL) techniques have gained significant traction and are widely utilized in implementing IDSs [13].

The widespread usage of Artificial Intelligence (AI) for generating IDSs makes them vulnerable to adversarial attacks [9]. These attacks involve maliciously crafted input data that deceive ML and DL models into making incorrect predictions or classifications. By exploiting vulnerabilities in the learning algorithms, adversaries can manipulate the behavior of IDSs and evade detection, potentially leading to system compromise or malfunction [19]. As a result, robust defenses against adversarial attacks are essential to ensure the reliability and security of vehicle systems. However, while extensive research has been conducted on adversarial attacks targeting IDSs of vehicle networks, these studies often require a set of assumptions that might not be realistic in real-world applications. As such, while the effectiveness of these attacks is alarmingly high in most studies, their threat model might not reflect the capabilities of actual attackers. Additionally, while some countermeasures have been proposed in the literature, their practicality and effectiveness against varied attacks are unclear.

Contribution. To address this gap in the literature on the practicality of adversarial attacks towards vehicle networks IDSs, we

present **CANEDERLI**, the first framework for evaluating the impact of transferability and adversarial training in the context of CAN-based IDSs. Our framework includes several model architectures and state-of-the-art attacks, allowing for a comprehensive evaluation of the adversarial impact. In the pursuit of a realistic threat model, our attacks are generated in white-box and black-box scenarios, reflecting the capabilities of real-world attackers. We incorporate adversarial training methodology based on fine-tuning procedures. Additionally, we introduce *adaptive online adversarial training*, which surpasses traditional techniques by preserving high accuracy and F1 score even during attacks, all while ensuring that the baseline performance of the models remains uncompromised. By making our framework open-source, we allow for the complete customization of models and attacks, allowing researchers and practitioners to evaluate better the resilience of their IDSs. Our contributions can be summarized as follows.

- We propose a novel framework for evaluating the impact of adversarial attacks and securing IDSs.
- We propose an adversarial training technique outclassing fine-tuning-based methodologies.
- We evaluate our system on a real-world dataset. Our evaluation includes multiple state-of-the-art attacks and model architectures.
- We open-source our code at: https://github.com/Mhackiori/CANEDERLI.

Organization. The paper is organized as follows. We analyze the literature on IDSs and their attacks in Section 2. Our system and threat model are detailed in Section 3. Section 4 delves into the methodology of our framework and the technical details of our contributions. We evaluate our framework in Section 5 and provide valuable takeaways from our study in Section 6. Finally, Section 7 concludes our work.

2 RELATED WORKS

Several IDSs have been recently proposed in the literature for vehicle networks. This section focuses on data-driven approaches that leverage ML and DL models. Starting from traditional on-road vehicles, the focus of IDSs has been the CAN bus. This protocol was developed by Robert Bosch GmbH in the 1980s and has quickly become a standard in internal vehicle networks due to its convenience for safety-critical applications [11]. Most techniques for detecting intrusions or attacks in the CAN bus involve ML models, frequency-based methods, statistical-based methods, or hybrid approaches [16]. For example, Kang et al. were the first to utilize a semi-supervised Deep Neural Network (DNN) for this purpose [12]. While some works propose the use of simple linear networks [6], others use more complex models such as Convolutional Neural Networks and Long Short Term Memory (LSTMs) [22]. Other approaches instead utilize the causality between data samples to predict the next value in a given sequence and evaluate divergence from the prediction [3]. Traffic frequency and statistics have also been used for this scope. Indeed, authors could obtain high accuracy in detecting anomalies by analyzing the behavior of interacting ECUs and extracting statistical properties of traffic time series [21]. The combination of these approaches yields the best results in the most varied scenarios, as ML models can leverage statistical and frequency data as features. This also allows for the real-time deployment of these systems and the online processing of the traffic [24]. With the increased connectivity of vehicles with other vehicles or infrastructures, the scope of IDSs has expanded to consider also V2V and V2I [2]. As such, while IDSs can still be mounted on the internal networks, they need to consider additional threats from the external environment. An even more challenging scenario is represented by Autonomous Vehicles (AVs), which, being equipped with several sensors and actuators, require heightened connectivity for their safe deployment.

While advantageous in terms of accuracy and complexity, the usage of ML and DL models for intrusion detection makes them vulnerable to adversarial attacks. Adversarial attacks involve crafting malicious input data to deceive AI models into making incorrect predictions or classifications, potentially leading to system compromise or malfunction. One class of adversarial techniques is evasion attacks, which entail crafting specific perturbations to input data to induce misclassification in the target model [9]. Another class is poisoning attacks, which manipulate the training data to compromise the model's performance at test time. While the real-world implementation of these attacks requires a set of assumptions on the attacker's capability, their application towards vehicle-based IDSs has been studied in the literature [4]. However, given the restricted threat model of attacks toward vehicles, one property that needs investigation is adversarial transferability, i.e., the capability of adversarial examples crafted for one model to fool another [1]. While the properties of these attacks have been partly studied in the literature [26], their real-world impact and consequences remain uncertain. This lack of clarity complicates the formulation of effective defenses against such attacks.

3 SYSTEM AND THREAT MODEL

We now discuss the assumptions that characterize the system functionality and the attacker's capability.

System Model. IDSs require access to the vehicle network traffic to operate. As such, the most straightforward implementation of the system is as an ECU connected directly to the CAN bus. In this scenario, access to the encoding and decoding schema of the vehicle packets is not required, as intrusion detection can be performed at the bit level. Furthermore, ECU IDSs can act as a filter and thus discard or flag malicious messages. Alternatively, IDSs can be implemented in the cloud. In this case, the vehicle should send the raw network traffic through an active external connection. This last scenario also allows for implementing more complex model architectures, as ECUs might have limited resources available for computation. Regardless of the implementation details, the IDS is constituted by a ML or DL model, taking in input single packet samples (or time windows, if using causal models) and producing an output flagging the packets as legitimate or malicious. In the case of abnormal packets, the model can also discern what type of intrusion it detects.

Threat Model. In real-world scenarios, an attacker might aim to compromise the security and safety of a vehicle without being detected by the IDS. For example, the CAN bus is notoriously vulnerable to Denial of Service (DoS) attacks, which can cause severe

incidents if not promptly addressed [18]. As such, we assume the attacker can read the vehicle network traffic and inject messages into the bus. This can be done through physical access to the On-Board Diagnostic (OBD) port or compromised connection toward other vehicles or infrastructures. When injecting malicious data, the attacker must apply perturbations to the attack packets. This perturbation needs to be substantial enough to avoid detection by the IDS but not too noticeable, as it might nullify the effect of the attack. However, in real-world scenarios, attackers cannot access the target model. This implies not being able to use its parameters and gradient for white-box attacks or using it as an oracle for crafting black-box attacks. Furthermore, training and validation datasets are kept private from the IDS manufacturers to prevent poisoning attacks. Thus, the attacker knows the classes (i.e., the attacks) labeled by the model but cannot access the source of the training data or its statistical distribution. We formalize three scenarios to comprehensively study the attacker's capabilities under different assumptions.

- *White-Box (WB)*: the attacker can access the target vehicle data and target model.
- *Gray-Box (GB)*: the attacker can access either the target vehicle data or the target model.
- *Black-Box (BB)*: the attacker cannot access the target vehicle data or the target model.

In the gray-box and black-box scenarios, the attacker can still perform evasion attacks by using surrogate models. These models have different architectures or are trained on other datasets. However, these attacks will be effective only if they present high transferability properties.

4 METHODOLOGY

We now delve into the methodology of CANEDERLI, our adversarial transferability and training framework. We discuss the models we use for testing and their parameters in Section 4.1. Section 4.2 overviews the evasion attacks we employ in our system. In Section 4.3, we show the adversarial training methods we use to defend our models and propose our novel technique. An overview of our framework is shown in Figure 1.

4.1 Models

We develop three models to evaluate the effectiveness and resilience of different DL architectures acting as an IDS: a linear DNN, a CNN, and an LSTM.

DNN. We utilize the linear DNN due to its simplicity and effectiveness in capturing linear relationships within the data, making it suitable for straightforward classification tasks. The model consists of two fully connected layers with ReLU activation functions in between. The input size is specified by the number of features we use (which will be discussed in Section 5.1), and the hidden layer size is set to 64.

CNN. We choose the CNN architecture for its ability to extract spatial features from data, which may reveal patterns indicative of intrusion activities in the network traffic. Its architecture incorporates a convolutional layer followed by a max-pooling layer. The input is a 1D signal, and the convolutional layer is defined with a kernel size of 3 and 32 output channels. After the convolution and pooling operations, the output is flattened and fed into a fully connected layer, which produces the final classification scores.

LSTM. We select the LSTM architecture to leverage its capacity to capture temporal dependencies and causal relationships within sequential data, which aligns well with the time-series nature of traffic data in a vehicular network. The model utilizes an LSTM layer for sequential data processing. It accepts input sequences with a length specified by the number of features and generates hidden states with a size of 64. The final classification is performed using a fully connected layer applied to the output of the last LSTM time step.

4.2 Attacks

In line with our threat model, we focus on evasion attacks since poisoning attacks are impossible without access to the training dataset. By defining x as the original sample, malicious samples can be created as $x^* = x + r$, where r is the perturbation. Most evasion attacks craft r through an optimization process similar to the following.

$$r = \arg\min_z f(x+z) \neq f(x). \tag{1}$$

In this equation, z is the variable under optimization, representing the perturbation added to the initial input x to yield the perturbed input $x + z$. We generate adversarial samples on the test set with Torchattacks [14], a popular Python library for crafting different evasion attacks [25]. We focus on the following types of attacks.

- *Basic Iterative Method (BIM)* – BIM is an iterative adversarial attack method that perturbs input data in small steps toward maximizing the model's loss. It aims to generate adversarial examples by iteratively applying small perturbations to input features [15].
- *Fast Gradient Sign Method (FGSM)* – FGSM is a single-step adversarial attack method that computes the gradient of the loss function with respect to the input data and perturbs the input in the direction of the gradient sign. Despite its simplicity, FGSM often produces effective adversarial examples [9].
- *Projected Gradient Descent (PGD)* – PGD is an iterative variant of the FGSM attack where the perturbations are constrained within a specified epsilon ball around the original data. By iteratively applying small perturbations while projecting them back onto the epsilon ball, PGD aims to generate strong adversarial examples [17].
- *Randomized Fast Gradient Sign Method* – RFGSM is a variant of the FGSM attack that introduces randomness in the perturbation process. By adding random noise to the perturbation direction, RFGSM aims to enhance the transferability and robustness of adversarial examples [23].

As anticipated in Section 3, the attacker needs to carefully scale the perturbation to be applied to the sample to avoid drastically modifying the packet values. For instance, in the case of the FGSM attack, malicious samples are generated as follows.

$$x^* = x + \varepsilon \cdot sign(\nabla_x J(\theta, x, y)). \tag{2}$$

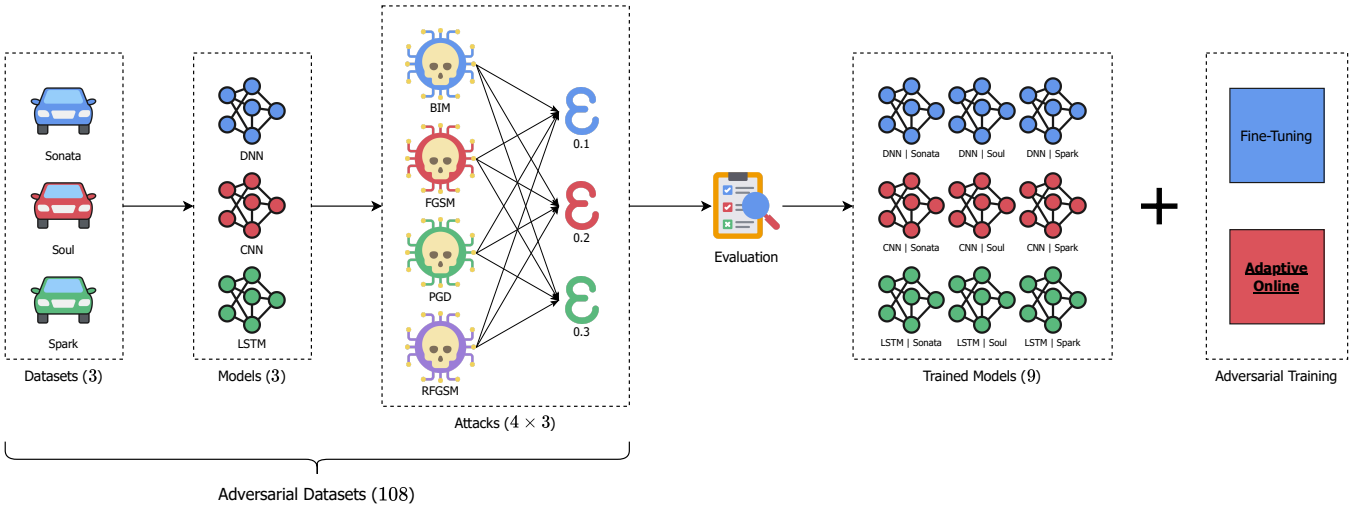

Figure 1: Framework overview.

Here, ϵ is the scaling factor, J is the loss function, θ represents the model parameters, and y is the ground truth for the input x. Higher ϵ can yield a higher Attack Success Rate (ASR), while lower ϵ reduce the perceptibility of the attack. To provide a comprehensive analysis of this tradeoff, we generate all attacks at three different maximum ϵ values: 0.1, 0.2, and 0.3

4.3 Adversarial Training

One of the most effective approaches for defending against adversarial attacks is adversarial training [5]. This procedure involves including adversarial attacks at training time to enhance the model's robustness. We use two different techniques for adversarial training. The first involves fine-tuning the pre-trained model on adversarial samples. While this approach has the advantage of being suitable to any trained model, it has the drawback of losing baseline performance. We propose *adaptive online adversarial training* to fix this issue. For each training epoch and batch, we first train the model on the legitimate inputs, then generate each attack in the same batch, evaluate them on the model, and backpropagate the loss (*"online"* as attacks are dynamically generated at each epoch) [7]. Furthermore, to reflect the increasing capabilities of the model as training commences, each attack is scaled with an increasing ϵ (*"adaptive"* as ϵ adapts to the model training). In particular, at each epoch i, attacks are scaled as follows.

$$\epsilon_i = \epsilon_{max} \frac{i}{num\_batches}. \tag{3}$$

This dynamic training approach confronts the model with progressively challenging adversarial examples, compelling continuous enhancement of its robustness.

5 EVALUATION

We now provide an experimental evaluation of our framework. First, in Section 5.1, we give details on the used dataset and the extracted features. In Section 5.2, we evaluate our models in an adversary-free scenario, while in Section 5.3, we evaluate the efficacy of our attacks. Section 5.4 evaluates and compares the proposed adversarial training techniques.

5.1 Dataset

For the evaluation, we use the Survival dataset [10]. This dataset focuses on three attack scenarios drastically affecting vehicle functions.

- *Flooding* – Being a multi-master network, the CAN bus manages collisions through arbitration. Thus, messages with higher priority (i.e., lower ID values) can overwrite packets with lower priority. This opens the possibility of DoS attacks, where attackers inject messages with low ID values to void the vehicle's functionalities.
- *Fuzzy* – Fuzzing is a testing technique used in software development to find vulnerabilities or bugs by injecting random or anomalous inputs. This attack uses the same principle for malicious purposes by injecting packets with random IDs.
- *Malfunction* – This attack focuses on specific IDs and overwriting their payload with different values from the original. This has the effect of abnormal vehicle behavior.

The dataset was collected by performing these attacks in three vehicles: HYUNDAI YF Sonata, KIA Soul, and CHEVROLET Spark. This produces three different datasets, one for each vehicle. The collected traffic comprises CAN bus packets containing their timestamp, ID, payload, and Data Length Code (DLC). We pre-process the dataset by converting the payload from hexadecimal to binary, thus increasing the number of features. Furthermore, we use the timestamp to compute intervals between messages with the same ID. This way, detecting injected messages becomes more straightforward with constant upload speeds. We label each packet with four possible values (three attacks and legitimate traffic). Finally, we balance our dataset with undersampling to avoid bias due to the data distribution. This ensures that, for each dataset, the number of packets for each label is the same.

5.2 Baseline Evaluation

We now evaluate the baseline performance of our IDSs. Since we tune our model hyperparameters offline, we divide our dataset in training (80% of the dataset) and testing (20% of the dataset). We train each model on each dataset, obtaining $3 \times 3 = 9$ trained models. Each model is trained for 30 epochs, using Adam as the optimizer with a learning rate of 0.001 and using cross entropy as the loss function. Results are shown in Table 1. All models obtain results close to perfection on all tasks. While some datasets appear to be easier than others (e.g., Sonata), accuracy and F1 score are high enough on all vehicles to provide low false positive and false negative rates.

Table 1: Baseline performance of the models.

Model	Sonata		Soul		Spark	
	Acc	F1	Acc	F1	Acc	F1
DNN	1.000	1.000	0.995	0.995	0.997	0.997
CNN	0.999	0.999	0.993	0.993	0.997	0.997
LSTM	1.000	1.000	0.995	0.995	0.995	0.995

5.3 Attacks Evaluation

To assess the effectiveness of our attacks, we evaluate the accuracy and F1 scores of the models when tested on adversarial datasets. Inspired by [1], we generate samples for each attack at each ϵ value for each model, obtaining $4 \times 3 \times 9 = 108$ adversarial datasets. Each of them is then evaluated on each model, obtaining $108 \times 9 = 972$ evaluations. We show the results in Table 2, where we split our evaluation based on the scenarios detailed in Section 3. The mean F1 score drop in all scenarios is significant as these attacks can completely disrupt the system's functionality. Furthermore, we notice strong transferability properties as attacks in the gray-box and black-box scenarios are also effective. This is due to the presence of attacks generated on different vehicle types, of which the characteristics are unknown to the legitimate system. We notice that the CNN model appears to be the most resilient. Instead, the types of attacks don't influence the score as much (Table 3).

Table 2: Average model' performance on adversarial datasets.

Model	F1 Score		
	WB	GB	BB
DNN	0.257	0.253	0.246
CNN	0.462	0.440	0.374
LSTM	0.254	0.294	0.283

Table 3: Average attacks' performance.

Attack	F1 Score		
	WB	GB	BB
BIM	0.362	0.380	0.356
FGSM	0.295	0.284	0.251
PGD	0.285	0.293	0.265
RFGSM	0.356	0.361	0.333

5.4 Adversarial Training Evaluation

To defend against these threats, the best course of action is performing adversarial training on the target model. First, we evaluate a traditional adversarial learning paradigm, consisting of fine-tuning the pre-trained model on an adversarial dataset. Secondly, we evaluate our adaptive online adversarial learning technique. The results are shown in Table 4 and Table 5. Fine-tuning-based adversarial learning methodologies can effectively defend against white-box attacks. However, one major drawback makes implementing this technique impossible in real-world scenarios: significant drops in baseline performance. As such, adversarially trained models lose their original accuracy and cannot perform in legitimate scenarios. Instead, our proposed online adversarial learning method shows similar F1 scores when under attack but maintains the baseline performance of the original models. Thus, our method balances performance and resilience against adversarial attacks.

Table 4: Fine-tuning-based adversarial training performance.

Model	Clean	F1 Score		
		WB	GB	BB
DNN	0.376	0.956	0.808	0.679
CNN	0.559	0.997	0.782	0.692
LSTM	0.367	0.963	0.804	0.661

Table 5: Adaptive online adversarial training performance.

Model	Clean	F1 Score		
		WB	GB	BB
DNN	0.991	0.936	0.796	0.671
CNN	0.996	0.880	0.741	0.621
LSTM	0.998	0.941	0.808	0.673

6 TAKEAWAYS

Our evaluation underscores the threat that evasion attacks pose to CAN-based IDSs. Thus, we now discuss and summarize the main takeaway messages we identify for proposing secure implementations in real-world scenarios.

Takeaway 1 – *A detailed definition of the system and threat models is necessary for thoroughly evaluating the scope of the threat.*

Different implementations of an IDS involve different assumptions on its functioning and the adversary knowledge. As highlighted by our results, the effect of attacks and defense measures highly depends on the application scenarios. Thus, knowing the attackers' capabilities when designing the system can significantly improve the effectiveness of the implemented countermeasures.

Takeaway 2 – *Black-box attacks can be as threatening as white-box attacks, as using surrogate models is an effective solution for the attacker.*

While models behave differently based on their architectures and the datasets they are trained on, attack behavior is consistent across different scenarios. For example, when performing a DoS

attack, the best strategy is to flood the traffic with packets with low IDs. As such, while regular traffic has different properties based on the encoding and the ECUs that constitute the vehicle, IDSs are trained to identify specific patterns used when also other vehicles are under attack. This makes this class of evasion attacks highly transferable.

Takeaway 3 – *IDSs security should be tackled during the design process, as adversarial fine-tuning strategies might not be efficient.*

Even though IDSs are designed for security purposes, their security is paramount for ensuring their effectiveness. As such, efficient adversarial training techniques are essential to defend against white-box and black-box evasion attacks. However, as our analysis shows, fine-tuning techniques drastically lower their accuracy in baseline performance. Therefore, our proposed adaptive online adversarial training is preferable. A detailed definition of the threat model is necessary to effectively train the models, as knowing what attacks the models are most probably encountering can increase their accuracy both in baseline performance and under attack.

7 CONCLUSIONS

Modern vehicles rely on numerous ECUs and robust communication through the CAN bus. Furthermore, communication extends to external entities such as other vehicles and infrastructures. However, it also introduces security risks, necessitating the implementation of IDSs. These systems usually leverage data-driven approaches, making them vulnerable to adversarial attacks. Unfortunately, the current literature on these attacks often lacks realistic assumptions and effective countermeasures, underscoring the need for further investigation and solutions.

Contribution. CANEDERLI addresses the gap in research on adversarial attacks on vehicle network IDSs. We introduce a framework for evaluating transferability and adversarial training impact, incorporating diverse model architectures and attacks. Our framework ensures realism by considering white-box and black-box scenarios, offering adaptive online adversarial training, and preserving model performance under attack. We open-source our framework for customization and IDS resilience assessment.

Future Works. Future research directions include conducting granular evaluations by examining various epsilon values in evasion attacks. Furthermore, more intricate model architectures and diverse datasets are worth exploring. This can also include the addition of diverse intrusion methodologies and attack techniques. These efforts aim to improve the resilience of IDSs against sophisticated adversarial threats, ensuring robust protection in dynamic environments.

REFERENCES

[1] Marco Alecci, Mauro Conti, Francesco Marchiori, Luca Martinelli, and Luca Pajola. 2023. Your Attack Is Too DUMB: Formalizing Attacker Scenarios for Adversarial Transferability. In *Proceedings of the 26th International Symposium on Research in Attacks, Intrusions and Defenses*. 315–329.

[2] Moayad Aloqaily, Safa Otoum, Ismaeel Al Ridhawi, and Yaser Jararweh. 2019. An intrusion detection system for connected vehicles in smart cities. *Ad Hoc Networks* 90 (2019), 101842.

[3] Aneetha Avalappampatty Sivasamy, Bose Sundan, et al. 2015. A dynamic intrusion detection system based on multivariate Hotelling's T 2 statistics approach for network environments. *The Scientific World Journal* 2015 (2015).

[4] Md Ahsan Ayub, William A Johnson, Douglas A Talbert, and Ambareen Siraj. 2020. Model evasion attack on intrusion detection systems using adversarial machine learning. In *2020 54th annual conference on information sciences and systems (CISS)*. IEEE, 1–6.

[5] Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. 2021. Recent advances in adversarial training for adversarial robustness. *arXiv preprint arXiv:2102.01356* (2021).

[6] Raghavendra Chalapathy and Sanjay Chawla. 2019. Deep learning for anomaly detection: A survey. *arXiv preprint arXiv:1901.03407* (2019).

[7] Emad Efatinasab, Francesco Marchiori, Alessandro Brighente, Mirco Rampazzo, and Mauro Conti. 2024. FaultGuard: A Generative Approach to Resilient Fault Prediction in Smart Electrical Grids. arXiv:2403.17494 [cs.CR]

[8] Mohamed El Zorkany, Ahmed Yasser, and Ahmed I Galal. 2020. Vehicle to vehicle "V2V" communication: scope, importance, challenges, research directions and future. *The Open Transportation Journal* 14, 1 (2020).

[9] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572* (2014).

[10] 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.

[11] International Standard Organization. 2015. *ISO 11898:2015: Road vehicles — Controller area network (CAN).* Standard. International Organization for Standardization, Geneva, CH.

[12] Min-Joo Kang and Je-Won Kang. 2016. Intrusion detection system using deep neural network for in-vehicle network security. *PloS one* 11, 6 (2016), e0155781.

[13] Ansam Khraisat, Iqbal Gondal, Peter Vamplew, and Joarder Kamruzzaman. 2019. Survey of intrusion detection systems: techniques, datasets and challenges. *Cybersecurity* 2, 1 (2019), 1–22.

[14] Hoki Kim. 2020. Torchattacks: A pytorch repository for adversarial attacks. *arXiv preprint arXiv:2010.01950* (2020).

[15] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. 2018. Adversarial examples in the physical world. In *Artificial intelligence safety and security*. Chapman and Hall/CRC, 99–112.

[16] Siti-Farhana Lokman, Abu Talib Othman, and Muhammad-Husaini Abu-Bakar. 2019. Intrusion detection system for automotive Controller Area Network (CAN) bus system: a review. *EURASIP Journal on Wireless Communications and Networking* 2019 (2019), 1–17.

[17] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. 2017. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083* (2017).

[18] Andrea Palanca, Eric Evenchick, Federico Maggi, and Stefano Zanero. 2017. A stealth, selective, link-layer denial-of-service attack against automotive networks. In *Detection of Intrusions and Malware, and Vulnerability Assessment: 14th International Conference, DIMVA 2017, Bonn, Germany, July 6-7, 2017, Proceedings 14*. Springer, 185–206.

[19] Han Qiu, Tian Dong, Tianwei Zhang, Jialiang Lu, Gerard Memmi, and Meikang Qiu. 2020. Adversarial attacks against network intrusion detection in IoT systems. *IEEE Internet of Things Journal* 8, 13 (2020), 10327–10335.

[20] Gopi Krishnan Rajbahadur, Andrew J Malton, Andrew Walenstein, and Ahmed E Hassan. 2018. A survey of anomaly detection for connected vehicle cybersecurity and safety. In *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 421–426.

[21] Hyun Min Song, Ha Rang Kim, and Huy Kang 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)*. IEEE, 63–68.

[22] Shahroz Tariq, Sangyup Lee, and Simon S Woo. 2020. CANTransfer: Transfer learning based intrusion detection on a controller area network using convolutional LSTM network. In *Proceedings of the 35th annual ACM symposium on applied computing*. 1048–1055.

[23] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. 2017. Ensemble adversarial training: Attacks and defenses. *arXiv preprint arXiv:1705.07204* (2017).

[24] Marc Weber, Simon Klug, Eric Sax, and Bastian Zimmer. 2018. Embedded hybrid anomaly detection for automotive CAN communication. In *9th European congress on embedded real time software and systems (ERTS 2018)*.

[25] Ruoyu Wu, Taegyu Kim, Dave Jing Tian, Antonio Bianchi, and Dongyan Xu. 2022. {DnD}: A {Cross-Architecture} Deep Neural Network Decompiler. In *31st USENIX Security Symposium (USENIX Security 22)*. 2135–2152.

[26] Ivo Zenden, Han Wang, Alfonso Iacovazzi, Arash Vahidi, Rolf Blom, and Shahid Raza. 2023. On the Resilience of Machine Learning-Based IDS for Automotive Networks. In *2023 IEEE Vehicular Networking Conference (VNC)*. IEEE, 239–246.