POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Executive Summary of the Thesis

# Exploring gradient-based evasion techniques against automotive intrusion detection systems

Laurea Magistrale in Computer Science and Engineering - Ingegneria Informatica

**Author:** Paolo Cerracchio

**Advisor:** Stefano Longari

**Co-advisor:** Michele Carminati

**Academic year:** 2022-2023

## 1. Introduction

Modern vehicles, particularly electric cars or autonomous vehicles, contain an ever-increasing number of electronic control units (ECUs) managing a myriad of features: from fuel injection to park assist, from adaptive cruise control to infotainment. These devices exchange data among each other through the controller area network (CAN) bus, the *de facto* worldwide standard for intra-vehicle communication. Moreover, cars nowadays communicate with the outside world through all sorts of technology, for example Bluetooth, GPS, USB and 3G, exposing numerous external access points.

In the last two decades, researchers have raised security concerns, culminating in the 2014 remote "hacking" demonstration of a Jeep Cherokee by Miller and Valasek. The experiment brought manufacturers to focus their resources in securing the CAN bus, since it uses a broadcast protocol devoid of any authentication or encryption mechanism, resulting in the widespread design and deployment of intrusion detection systems (IDSs). An automotive IDS is usually a piece of software that monitors all the messages exchanged on the network to detect any anomalous pattern or behaviour. The most common detection methods involve analysis of packet arrival times, succession of IDs or payload-based anomaly detection through statistical or deep learning (DL) models. While they are widely used for their effectiveness and generalization capabilities, it is well-known that DL models are vulnerable to *adversarial examples*: inputs specifically perturbed to induce an error in a target discriminator and designed to not be recognized as such upon human inspection.

Current research on the adversarial problem is focused on computer vision, malware recognition and network security applications, therefore very few specialized studies exist in the automotive domain. The robustness of automotive IDSs is crucial, my goal in the present work is to test relevant detection architectures against a knowledgeable adversarial attacker. My main contributions are the following:

- I benchmark six different IDSs designs against two publicly available datasets;
- I design, implement, tune and test three custom variants of the popular basic iterative method (BIM) [1] and *DeepFool* [3] evasive perturbation algorithms, adjusted to work with CAN packet signals;
- I perform additional transferability and

reusability tests for these newly introduced adversarial techniques, in order to explore the feasibility of this kind of attack under different conditions.

## 2. Background

**Adversarial evasion** In DL, evasion is the act of finding a subtle perturbation to cause sample misclassification; it can be formalized as the optimization problem of finding the adversarial example $\tilde{x}$ undetected by the discriminator $F(x)$ while minimizing the perturbation $\delta(x^*, x)$:

$$\begin{aligned} \tilde{x} = argmin_{x^*} [\delta(x^*, x)] \\ \text{s.t.} \quad F(x^*) = 0 \end{aligned} \tag{1}$$

The earliest and most common type of adversarial algorithms are gradient-based evasion algorithms, such as the fast gradient method (FGM) and the already mentioned BIM and *DeepFool*. While they may differ in approximations, assumptions and distance metrics, the rationale behind these approaches is to leverage the back-propagation algorithm against the model under attack. The idea is to push the original input towards areas in the problem space where the classifier confidence is lower, approaching and eventually crossing the decision boundary by exploiting gradient ascent of an arbitrary loss function (oftentimes the same used during the training of the model).

As it is uncommon for a malicious actor to have access to the model, researchers developed black-box techniques that try to fool a victim via iterative queries. However, all the aforementioned adversarial attacks are tailored towards image pixel perturbation, needing a greater effort and further analysis to be applied to tabular, heterogeneous and otherwise constrained data. In particular, *Noseda* [4] develops a black-box greedy evasion strategy and tests it against automotive IDS, albeit the ensuing evaluation lacks a suitable method to quantify the semantic distortion introduced by the proposed attack.

**Automotive IDSs** In the automotive field we distinguish two main detection paradigms: flow-based detection, usually leveraging lightweight methods such as distribution fitting, regression or convolutional neural networks to detect intrusion through timestamps, order of IDs, hamming distance and other heuristics on the traffic, and payload-based detection, usually employing more complex DL techniques like autoencoders. The former can easily detect attacks that add new messages to the traffic, regardless of their content, while it struggles against impersonation spoofing attacks, that can be detected by the latter. *Taylor et al.* [5] propose a long short-term memory (LSTM) predictor that, given a sequence of packet payloads, tries to predict the next one, the distance between the guess and the actual upcoming packet gives an *anomaly score*. *Longari et al.* [2] design CANnolo, an LSTM-based autoencoder that instead tries to reconstruct the whole sequence of payloads to obtain a similar score; their work also tackles the problem of CAN bus semantics, as the manufacturers attempt to enhance the security of ECUs through obscurity, keeping secret the application-level layout of the messages.

**Motivation** In the swiftly evolving automotive field, new security challenges arise as the industry paves the road for innovation. While the research tries to design future-proof defenses, they also uncover the menace of evasion, threatening state-of-the-art detectors. In order to address this issue, the purpose of my thesis is to explore sensible adversarial attacks in this specific setting. More precisely, my interest is to analyze the response of modern IDSs in the worst case scenario with a capable and knowledgeable attacker, to better understand the robustness and characteristics of the architectures under test in such a scenario. From an offensive standpoint, the possibility to algorithmically produce evasive examples has an undeniable appeal and would represent a critical vulnerability in DL models.

**Threat Model** We consider a single CAN bus monitored by a network IDS, the goal of the attacker is to inject an objective sequence of malicious packets to achieve a range of effects while avoiding detection. The attacker has full white-box access to the IDS, this includes the set of input features, the structure, a representative training dataset and all the parameters; she can also compromise different ECUs on the bus. Thus, given the lack of native security mechanisms, she can inject arbitrary messages at any given instant and can intercept inbound and outbound communications, effectively eavesdropping the whole traffic on the channel. The adversarial strategy consists in perturbing a predetermined set of malicious sequences with the proposed algorithms, trying to morph existing samples into evasive examples. For the purpose of our exploration, we establish the following assumptions:

1. All real-time constraints typical of the automotive domain are relaxed, the algorithms introduced in this work query multiple times the same DL models proposed for detection, hence the expected time required to perform all the computation needed by the attack is up to two order of magnitudes greater than the corresponding IDS inference time;
2. The attacker has complete control over the ECUs he compromises and can impersonate it, this is reasonable to assume given the wide attack surface and the established methods to silence an ECU;
3. Since car manufacturers are usually very secretive about the semantics of CAN messages, we will exploit reverse engineering methods to extract signals in the traffic and treat them as the reliable intended values.

## 3. Methodology

**IDS architecture and models** The IDS I implement for this study is a payload-based anomaly detector leveraging different core models from the state-of-the-art. This type of unsupervised IDS is a binary classifier that decides whether the input is malicious or not. We feed the sample into a DL model that outputs either a reconstruction of the input, in the case of autoencoders, or a prediction of the next element in the sequence, in the case of predictors, we then compute the distance between the produced output with the actual target value using the mean squared error (MSE) function. This process yields an *anomaly score*: a high score identifies an outlier input. In practice, the actual decision compares the score against a threshold value, which can be identified through various strategies involving inference on a *thresholding set*. I evaluate the following models:

- *FFNN*: a simple one-to-one autoencoder with two fully connected layers, note that this architecture is blind to replay attacks;
- *CANnolo* [2]: a window-to-window symmetrical autoencoder with two fully connected and two LSTM layers;
- *LSTM-based predictors* [5]: two window-to-one predictor variants, I implement a short version with just two LSTM layers and a long version with four LSTM layers;
- *GRU-based predictors*: two window-to-one predictors analogous to the short and long LSTM variants, employing the more lightweight gated recurrent units (GRUs).

**Signal extraction and preprocessing** To identify the signals in the payloads from any CAN ID I implement the reverse engineering of automotive data frames (READ) method by Marchetti and Stabili, as explained in the CANnolo paper [2]. This method performs reverse engineering through the study of the rate at which every bit flips, allowing us to extract all and only the bit ranges corresponding to physical values or binary flags, eliminating constant bits, counters and CRCs. We normalize the obtained features to the $[0, 1]$ range by dividing each value by the maximum representable integer for the corresponding range.

**Adversarial algorithms** In the proposed scenario the attacker starts with an initial CAN log having some unperturbed injected frames as a baseline, then her actual strategy is to morph these intended sequences to be evasive by applying repeated perturbations, Figure 1 illustrates a single iteration of the process. To implement the scheme, I design three algorithms.

The **BIM-based algorithms** [1] iteratively apply the generalized FGM perturbation (2). The first variant, named *step decay* BIM, is identical to the base variant with $p = 1$, except for the step hyperparameter: it is not a constant value $\epsilon$ but rather a decaying value $\epsilon^{t+1} = \epsilon^t \cdot \omega$, where $\omega$ is an arbitrary decay rate, that I fix at 0.9 upon linear search. The second variant employs the $l2$ norm with $p = 2$, so that $\epsilon$ is the module of the perturbation and the direction precisely points towards the maximum of the loss function $L$. The rationale behind this type of attack is to approximate as linear the behaviour of the classifier around the current model parameters, in order to push the sample towards a maximum

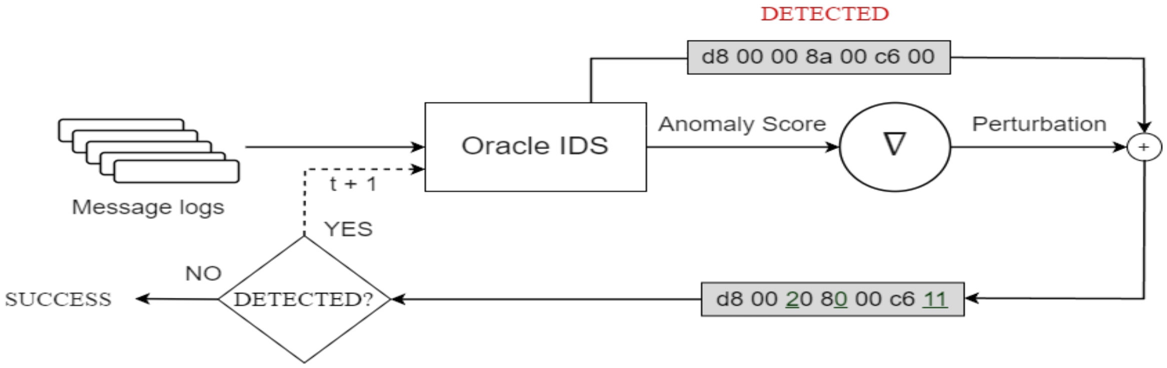

Figure 1: Single iteration view of the implemented evasion strategy

of the loss function (or, in our case, a minimum of the anomaly score).

$$x^{t+1} = clip_X(x^t - \epsilon \frac{\nabla_x L(w, x^t)}{||\nabla_x L(w, x^t)||_p}) \qquad (2)$$

The ***DeepFool*-based algorithm** [3] is also iterative. We compute the perturbation $\delta$ according to Equation (3); in the equation, $F(w, x)$ is the decision function with boundary at 0, to apply the method to the anomaly score $L$ I perform the simple translation $F(w, x) = L(w, x) - \theta_{IDS}$, where $\theta_{IDS}$ is the threshold value. This approach approximates the IDS as an affine classifier $w \cdot x + b = F(w, x)$, then the perturbation tries to push the sample beyond the affine decision boundary, overshooting by a factor equal to $1 + \epsilon$.

$$\delta^t = (1 + \epsilon) \cdot (-F(w, x) \frac{\nabla_x F(w, x^t)}{||\nabla_x F(w, x^t)||_2^2})) \qquad (3)$$

In either method, the *clip* function ensures that the resulting $x$ is a valid payload after each step by performing value clipping of the normalized signal in the $[0, 1]$ range and by rounding it to the closest integer representable within the corresponding bit range. If after 50 iterations the algorithms have not produced an adversarial example, we terminate the evasion attempt as failed.

## 4. Experimental Evaluation

**Datasets** I choose two datasets for the present study: the *C-1 ReCAN* dataset, containing real attack-free traffic from a Giulia Veloce car [2, 4], and the *car hacking (CH)* dataset, a popular benchmarking dataset containing both normal traffic and intrusion experiments from a real vehicle. To perform the experiments, I augment experiment 7 in the ReCAN dataset with *CANtack* [2], injecting the following baseline attacks:

- *Injection replay attack*: adds new packets to the flow of messages with a rate of 0.4, meaning that the malicious payloads are injected two and a half times slower than the average inter-arrival time of normal frames. The payloads are sniffed from previous legitimate sequences;
- *Full replay attack*: it is a masquerade attack, tampering with outbound packets to substitute payloads sniffed from previous legitimate sequences;
- *Continuous attacks*: while performing a replay masquerade attack, the attacker gradually brings a signal in the payloads to an arbitrary value. This is implemented in two variants, either with a random target or to the the final value of 0;
- *Fuzzy attack*: while performing a masquerade attack, the attacker randomizes all the bits belonging to signals in the payloads;

All the aforementioned attacks are organized in 10 non-overlapping sequences composed by 25 CAN frames each. Conversely, the CH dataset only contains two relevant attacks: a flood attack that spoofs two different CAN IDs and a sparse fuzzy attack across several ECUs at once. I select 12 IDs compatible with temporal-based analysis from ReCAN [2] and the two spoofed IDs from CH. It is important to note that I find the attack-free traffic dump in the CH to not be representative of the non tampered traffic distribution in other experiments, causing the tested IDSs to behave erratically, because of this reason and of the different attack design, it is only included in the main experiment.

**Main comparative experiment** The main experiment consists in cross-testing of the 6 models against the perturbation from the 3 proposed algorithms for all the available attack scenarios. I consider three metrics to evaluate each IDS: the standard area under the curve (AUC) and true positive rate (TPR), also called recall, and the aggregate perturbation (AP), a custom metric that represents the average maximum perturbation percentage of the signals in a packet. The results reveal that CANnolo is the most resilient model to the proposed white-box strategy, while the predictive models are all equally easier to evade. In particular, the greatest recall drop is respectively 8.9% for CANnolo and around 50% for the LSTM architectures, however, upon inspection of the perturbed samples I notice that, in many cases, the injected signals are more similar to the corresponding payload in the attack-free state than to the intended malicious content, losing the attack semantic, this corresponds to a high AP score. The algorithms are more efficient in perturbing “harder” samples, finding very close adversarial examples near the decision boundary, moreover, in some other cases, the required deviation from the normal behaviour is just too large or too abrupt for the method to succeed, so the injected sequence has just a couple of packets out of its 25 components that could not evade detection. In general, *DeepFool* tends to achieve high evasion rate at the cost of unacceptably high perturbation, the $l2$ BIM is often the preferable choice, while the *decay* variant is sometimes constrained to perturbations too small to effectively produce adversarial examples.

**Transferability experiment** In this grey-box experiment the attacker uses an *oracle* — a substitute model — against which he generates evasive input in place of an unknown detection model, in order to transfer the obtained adversarial attack on the actual deployed architecture. I choose the best performing algorithms for 3 oracles: an FFNN, CANnolo and a long LSTM. The most notable result is that the transfer of evasive examples from the LSTM to CANnolo is more successful than the simple white-box attack in all scenarios except the fuzzy one. Table 1 showcases this behaviour through the recall metric, I highlight in bold the best performing evasion strategy. Also note that in the injection replay and fuzzy scenarios the LSTM-based adversarial examples are much closer to the original packets than the ones produced with *DeepFool* and white-box access. I explain this phenomenon by the complexity and autoencoder structure of CANnolo, that cause the algorithms to require many iterations and hinder a steady convergence. Vice versa, samples produced against CANnolo do not reliably transfer to any other model, while the FFNN oracle achieves noticeable transfer rates on the continuous attacks, but has inferior performance in comparison to the LSTM.

**Precomputation experiment** In the precomputation scenario the attacker prepares an adversarial perturbation in advance and tries to reinject it at a later time. In practice, I consider all the sequences that succeed in evading classification as a whole; I also consider as candidate injection points every point in the traf-

| | Full Replay | Continuous | Continuous to Minimum | Fuzzy | Injection Replay |
|---|---|---|---|---|---|
| Baseline | 0.6615 | 0.9196 | 0.9267 | 1.0000 | 0.7273 |
| LSTM oracle $l2$ BIM | **0.5677** | **0.8543** | **0.8848** | 0.9895 | **0.5588** |
| White-box *DeepFool* | 0.5729 | 0.8844 | 0.9162 | **0.9842** | 0.6578 |

Table 1: Comparison of the baseline TPR of CANnolo with the TPRs against the best performing white-box algorithm and the LSTM transfer attack.

fic preceded by at least ten packets identical to the preamble of the original attack location. I exclude the FFNN model as it performs classification independently of the sequence of messages, and CANnolo as, given its superior resilience, there were not enough completely evasive sequence to carry out the test. Following this procedure, I find that such reuse of adversarial sequences has limited feasibility: out of 12 CAN IDs, only 2 of them were susceptible to this strategy, these ECUs have in common very slowly varying physical signals and the tendency to gravitate around some more frequent values, rather than having a more uniform distribution. These characteristics also justify the high number of candidate reinjection point found for every adversarial sequence, resulting in over 1000 successful new evasions for each one of them. In general, I observed that effective reinjection required a very close similarity between the reinjection point preamble and the original frames preceding the malicious message, with an average of 37 identical packets in a row relatively to the 39 packets long prediction window.

## 5. Conclusions

The objective of this thesis is to explore the behaviour of 6 IDS architectures from the state-of-the-art against the three proposed adversarial algorithm variants for the CAN bus. I mitigate the lack of proprietary information through reverse engineering of the signals in the traffic.

Overall, I find the proposed attacks to achieve high evasion rate against the predictive models while CANnolo proves to be more resilient; however, both the success and the quality of the adversarial examples highly depend on the starting payload. In particular, I find that more sophisticated attacks, taking into account signal intercorrelation, not only are more easily morphed to be evasive, but also produce higher quality adversarial points that remain closer to the intended content. Moreover, the transferability results show that, in some cases, attacking an oracle for better convergence could be more effective than even performing a white-box attack on a complex model such as CANnolo, under all the metrics discussed during the evaluation phase.

A limitation in my analysis is the unavailability of a cyberphysical system to evaluate the outcomes of various injections in a real-world setting. Given the discrepancies highlighted in the CH dataset, I consider of paramount importance establishing a well-defined benchmarking framework for IDS performance and adversarial resilience. In conclusion, in light of the impact the original attack has on the reviewed evasion strategy, I would be interested in extending this analysis to a combined strategy that would employ an advanced generative method alongside the proposed algorithms.

## References

[1] Alexey Kurakin, Ian J Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *Artificial intelligence safety and security*, pages 99–112. Chapman and Hall/CRC, 2018. From 2016 preprint.

[2] Stefano Longari, Alessandro Nichelini, Carlo Alberto Pozzoli, Michele Carminati, and Stefano Zanero. Candito: Improving payload-based detection of attacks on controller area networks. *arXiv preprint arXiv:2208.06628*, 2022.

[3] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[4] Francesco Noseda. Evasion attacks against intrusion detection systems on communication area network. Master's thesis, 2022.

[5] Adrian Taylor, Sylvain Leblanc, and Nathalie Japkowicz. Anomaly detection in automobile control network data with long short-term memory networks. In *2016 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pages 130–139. IEEE, 2016.