University of Padova

Department of Mathematics "Tullio Levi-Civita"

Master Thesis in Computer Science

GAN-CAN: A Novel Attack to Behavior-Based Driver Authentication Systems

Supervisor
Prof. Mauro Conti
University of Padova

Co-supervisor
Denis Donadel
University of Padova

Master Candidate
Emad Efatinasab

Student ID
2044422

Academic Year

2022-2023

Abstract

For many years, car keys have been the sole mean of authentication in vehicles. Whether the access control process is physical or wireless, entrusting the ownership of a vehicle to a single token is prone to stealing attempts. Modern vehicles equipped with the Controller Area Network (CAN) bus technology collects a wealth of sensor data in real-time, covering aspects such as the vehicle, environment, and driver. This data can be processed and analyzed to gain valuable insights and solutions for human behavior analysis. For this reason, many researchers started developing behavior-based authentication systems. Many Machine Learning (ML) and Deep Learning models (DL) have been explored for behavior-based driver authentication, but the emphasis on security has not been a primary focus in the design of these systems.

By collecting data in a moving vehicle, DL models can recognize patterns in the data and identify drivers based on their driving behavior. This can be used as an anti-theft system, as a thief would exhibit a different driving style compared to the vehicle owner. However, the assumption that an attacker cannot replicate the legitimate driver behavior falls under certain conditions.

In this thesis, we propose **GAN-CAN**, the first attack capable of fooling state-of-the-art behavior-based driver authentication systems in a vehicle. Based on the adversary's knowledge, we propose different GAN-CAN implementations. Our attack leverages the lack of security in the CAN bus to inject suitably designed time-series data to mimic the legitimate driver. Our malicious time series data is generated through the integration of a modified reinforcement learning technique with Generative Adversarial Networks (GANs) with adapted training process. Furthermore we conduct a thorough investigation into the safety implications of the injected values throughout the attack. This meticulous study is conducted to guarantee that the introduced values do not in any way undermine the safety of the vehicle and the individuals inside it. Also, we formalize a real-world implementation of a driver authentication system considering possible vulnerabilities and exploits. We tested GAN-CAN in an improved version of the most efficient driver behavior-based authentication model in the literature.

We prove that our attack can fool it with an attack success rate of up to 99%. We show how an attacker, without prior knowledge of the authentication system, can steal a car by deploying GAN-CAN in an off-the-shelf system in under 22 minutes. Moreover, by considering the safety importance of the injected values, we demonstrate that GAN-CAN can successfully deceive the authentication system without compromising the overall safety of the vehicle. This highlights the urgent need to address the security vulnerabilities present in behavior-based driver authentication systems. In the end, we suggest some possible countermeasures to the GAN-CAN attack.

Contents

Abstract v

List of figures ix

List of tables xi

Listing of acronyms xiii

1 Introduction 1
1.1 Behavior-Based Car Anti-Theft 2
1.2 Is Behavior-Based Authentication Really Secure? 3
1.3 Contributions 3
1.4 Organization 5

2 Background 7
2.1 CAN Bus and ECUs 7
2.2 Generative Adversarial Networks 9
2.3 Behavior-Based Authentication 10
2.4 Related Works 11
2.4.1 CAN bus security 11
2.4.2 Intrusion Detection Systems on CAN bus 12
2.4.3 Adversarial attacks against IDS 13

3 Behavior-Based Driver Identification System 15
3.1 Toward a Perfect Behavior-Based Driver Identification System 15
3.2 Authenticator System Model 18
3.2.1 Pipeline 18
3.2.2 System Model Scenarios 19
3.2.3 Authentication Device Implementation 21

4 Threat Model 25
4.1 White Box (WB): Known Model and Data 25
4.2 Gray Box 1 (GB1): Unknown Model, Known Data 26
4.3 Gray Box 2 (GB2): Known Model, Unknown Data 26
4.4 Black Box (BB): Unknown Model and Data 27

5 GAN-CAN Methodology 29
5.1 Features Safety . . . 29
5.2 Gray Box 1 Scenario . . . 30
5.3 Gray Box 2 Scenario . . . 32
5.4 Black Box Scenario . . . 35
5.4.1 Gaining knowledge on the data (BB1) . . . 35
5.4.2 Gaining knowledge on the model (BB2) . . . 35

6 Evaluation 37
6.1 Baseline Authentication Results . . . 38
6.2 GAN-CAN Results . . . 38
6.2.1 GB1 Evaluation . . . 38
6.2.2 GB2 Evaluation . . . 39
6.2.3 BB1 Evaluation . . . 41
6.2.4 BB2 Evaluation . . . 41
6.3 Summary . . . 43

7 Countermeasures 45
7.1 Does Behavior-Based Authentication alone provide an effective solution ? . . 46

8 Conclusion and future works 49
8.1 Conclusion . . . 49
8.2 Future works . . . 50
8.2.1 Refining and improving GAN-CAN . . . 50
8.2.2 Exploring countermeasures . . . 50
8.2.3 Assessing the impact on real-world scenarios . . . 50
8.2.4 Ethical considerations and responsible disclosures . . . 51
8.2.5 Testing with bigger datasets . . . 51

References 53

Acknowledgments 67

Listing of figures

1.1 The authenticator permits only legitimate drivers to run a vehicle. However, with **GAN-CAN**, an attacker can fool the authenticator and steal the vehicle by connecting a malicious device to the CAN bus. 4

2.1 The ECUs use the CAN interface for data communication [1]. 8
2.2 Classical CAN frame structure [2]. 9
2.3 Complete CAN bus frame [3]. 9
2.4 Traditional GAN Training Process. 10

3.1 Pipeline of the authenticator system. 19
3.2 A schema of the system and threat model. 20

5.1 Schema of the main steps for each attack. 36

6.1 ASR for each target driver and each attacker driver for the GB1 scenario. . . . 40
6.2 Episode of convergence of the generator model for each driver. 42

Listing of tables

3.1 Summary on the different models employed by papers in the literature. The paper's target can be to profile the driving *style* of the driver or to identify the *driver*. 16

4.1 Different cases based on the attacker's knowledge. 25
4.2 Which is the easiest attack that will work in a certain scenario? With *GAN-CAN, and then replay*, GAN-CAN is only needed to generate one sample to be submitted as a replay attack to the authenticator. 27

5.1 Distribution of features into safety classes. 31
5.2 Architecture of the Generator. 33

6.1 Baseline results of the authentication systems on the OCSLab dataset [4]. . . 38
6.2 Summary of the GAN-CAN results. 43

Listing of acronyms

AI	Artificial Intelligence
ASR	Attack Success Rate
CAN	Controller Area Network
CNN	Convolutional Neural Network
DDoS	Distributed Denial of Service
DNN	Deep Neural Network
ECU	Electronic Control Unit
EV	Electric Vehicle
GAN	Generative Adversarial Network
GPS	Global Positioning System
GRU	Gated Recurrent Unit
IDS	Intrusion Detection Systems
LSTM	Long Short-Term Memory
LSTM-FCN	Long Short-Term Memory Fully Convolutional Network
ML	Machine Learning
DL	Deep Learning
OBD	On-Boad Diagnostic
RNN	Recurrent Neural Network
SVDD	Support Vector Domain Description
SVM	Support Vector Machine
TPM	Trusted Platform Module

UBM	Universal Background Model
UBI	Usage Based Insurance
UDS	Unified Diagnostic Services
MitM	Man-in-the-Middle
RPM	Revolutions Per Minute
RF	Random Forest
GMM	Gaussian Mixture Model
kNN	k-Nearest Neighbors
GNN	Graph Neural Network
RWR	Random Walk With Restart
ECC	Elliptic Curve Cryptography
GGNB	Graph-based Gaussian naive Baye
ACGAN	Auxiliary Classifier Generative Adversarial Network
HAA	Hierarchical Adversarial Attack
NIDS	Network-based Intrusion Detection System

1

Introduction

Car theft is a problem that nowadays is not completely solved. Technology development provided solutions to make thieves' life harder, e.g., by turning a car from a simple ignition system to a cryptography-based co-presence detection [5]. However, recent news showed that it is still possible for attackers to steal cars, e.g., by directly connecting to the Controller Area Network (CAN) bus and injecting packets [6]. Numerous driver authentication methods have been proposed in the literature, encompassing various approaches such as all sort of biometric authentication techniques [7, 8, 9, 10, 11], behavior-based authentication, and more. These methods have been extensively studied to enhance the security and effectiveness of driver authentication systems. For instance Derman et al. [7] propose a deep neural network-based approach for real-time and continuous authentication of vehicle drivers. The study aims to assess the applicability of current face recognition technology for practical implementation. Gupta et al. [9] introduce DriverAuth, a risk-based multi-modal biometric authentication solution for enhancing safety in on-demand ride and ride-sharing services. DriverAuth employs three biometric modalities (swipe, text-independent voice, and face) to verify registered drivers. Current models utilize diverse modalities, leading to limitations in their applications or inconvenience for drivers [12]. Moreover, identification technologies like fingerprint recognition lack the capability to continuously monitor the driver's identity in real-time [13]. In light of these problems we will take a deeper look into behavior-based authentication systems.

1.1 Behavior-Based Car Anti-Theft

To solve the car theft problem, researchers proposed the use of driver behavior data as an additional test to verify the legitimacy of the driver [13, 14, 15]. Behavioral-based driver identification uses the driving style to distinguish among different drivers. To this aim, the authentication system collects Controller Area Network (CAN) bus data and extracts features peculiar to each driver from the time evolution of information such as location, speed, braking, and acceleration. Behavioral-based driver authentication systems' use of CAN bus data allows it to be easily implementable in modern vehicles with minimal costs. Traditional defense mechanisms such as key fobs or immobilizers have been proven to be ineffective in preventing unauthorized access to the vehicle since attackers can leverage several entry points to accomplish their purposes [6, 16]. Despite identifying a particular driver for authentication purposes, profiling user behaviors can also be used in other contexts. Recently, insurance companies have started implementing Usage Based Insurance (UBI) policies where individuals are expected to purchase an insurance plan commensurate to their driving behavior instead of offering a fixed pricing convention. With this system, drivers with aggressive driving behaviors are forced into a higher fee with respect to more responsible drivers [17]. Similarly, identification of drunk drivers can be important for safety reasons [18].

Thanks to the recent advancements in Artificial Intelligence (AI), most of the behavior-based authentication models leverage Machine Learning (ML) or Deep Learning (DL) techniques to verify the driver legitimacy [19, 20]. While some authors explore the use of simpler models (i.e., non-deep) for this task [21], the best accuracy results are usually obtained with Recurrent Neural Networks (RNNs) and Long Short Term Memory (LSTM) networks [22, 23]. Indeed, these kinds of models can grasp the causal relationship between data samples, uncovering the key characteristics of driver behavior. Thanks to the constant data feed from the CAN bus, identification can be performed in real-time and can constantly provide feedback on the legitimacy of the driver to the anti-theft system. It is worth noting that while the final goal of these systems is authentication, in the literature the technical problem is tackled as an identification task (i.e., datasets have multiple drivers, and the model is tasked with identifying each one).

1.2 Is Behavior-Based Authentication Really Secure?

The behavior-based anti-theft system works under the assumptions that: i) the attacker is not able to mimic the legitimate driver behavior, and ii) the attacker is not able to inject packets in the CAN bus without causing safety-threatening driving behaviors while stealing the car. Regarding the first assumption, given that driver identification is based on AI models, we see that such an assumption may fall apart since AI algorithms have been proven vulnerable to several types of attacks. Indeed, adversarial attacks, such as evasion [24] or poisoning [25] attacks, have been proven successful in affecting the behavior of classifiers. In the former, attackers leverage their knowledge of the model to generate specifically crafted perturbations that can be applied to data samples to drive the target model to misclassification. In the latter, attackers accessing the dataset during the training procedure can inject malicious samples that compromise the model's behavior. While most of these attacks include strong assumptions about the attacker's knowledge of the system, several works on attack transferability show that many real-world scenarios can still be targeted by them effectively [26]. Moreover, the usage of generative models such as Generative Adversarial Networks (GANs) is actively researched for both malicious and defensive purposes, given their ability to generate increasingly realistic data samples that can fool a classifier also in black-box scenarios [27].

Regarding the second assumption, i.e., the difficulty of injecting packets without hindering driving safety, we notice that not all packets exchanged in the CAN bus are related to safety-sensitive information. This opens a possibility for the attacker to inject maliciously crafted instructions. If the behavior-based anti-theft system were to use non-safety-sensitive information to assess whether the driver is the legitimate one, then the attacker could mimic the legitimate behavior without impacting driving safety while stealing the car. As discussed in Chapter 5.1 and also demonstrated in other works [15, 14], non-safety-sensitive features are the most important ones in characterizing a driver.

1.3 Contributions

In this thesis, we propose **GAN-CAN**, a novel attack framework able to fool behavior-based car anti-theft systems, as summarized in Figure 1.1. Our attack works in different scenarios based on the attacker's prior knowledge of the legitimate part of the system. We first consider a White Box (WB) scenario, where the attacker knows the authenticator model and has access to legitimate driver's data. We then consider two Gray Box (GB1 and GB2) scenarios, where

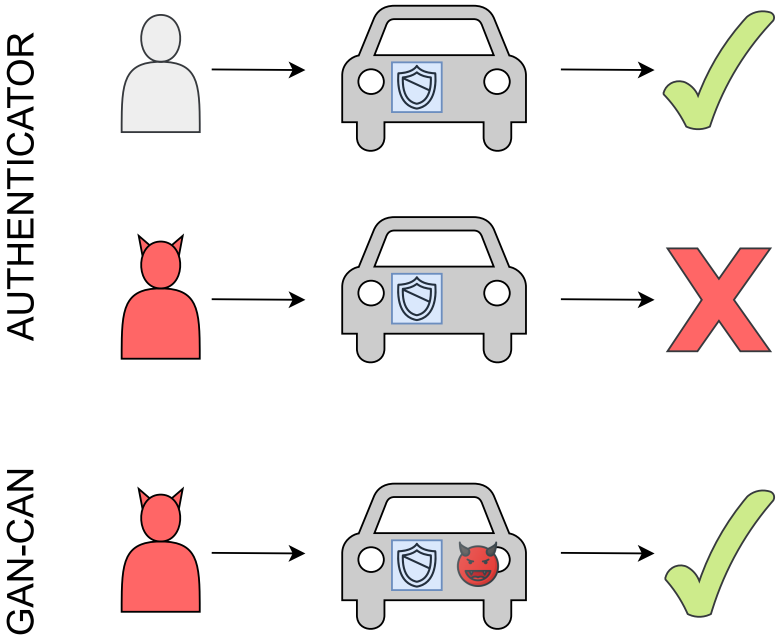


Figure 1.1: The authenticator permits only legitimate drivers to run a vehicle. However, with **GAN-CAN**, an attacker can fool the authenticator and steal the vehicle by connecting a malicious device to the CAN bus.

the attacker can either access the data or the model, respectively. Finally, we consider the Black Box (BB) scenario where the attacker does not know the model nor the legitimate data. We then study the features an attacker can safely inject without incurring in safety-threatening car state changes during the theft. Based on the considered scenarios, we design generative models based on the Generative Adversarial Network (GAN) framework to generate and inject malicious packets in the CAN bus to fool the identification system without affecting the core functionalities of the vehicle. GAN-CAN highlights several vulnerabilities in the possible implementations of behavioral biometrics for driver authentication and works even on the strictest assumptions. Indeed, we demonstrate that leveraging On-Boad Diagnostic (OBD)-II data to extract features for a driver identification system is an easily exploitable practice. Instead, even when using data directly from the CAN bus, our attack can overwrite a selected subset of packets to continuously authenticate an unauthorized user driving the target vehicle.

The main contributions of our work can be summarized as follows.

- We propose a model of the strongest behavior-based driver authentication system based on our research on the state of the art in behavior-based driver classification. We analyze the different solutions and compare them to stress their pros and cons.

- We present **GAN-CAN**, a novel attack on driver identification systems. Our attack injects specifically crafted malicious data into the CAN bus causing the authentication system to recognize the attacker as the legitimate driver without affecting the core functionalities of the vehicle. This makes GAN-CAN a practical and implementable attack.

- We formalize a real-world implementation of a driver identification system taking into account possible vulnerabilities and exploits. We show that the data collection procedure used in several works (i.e., , OBD-II protocol) is vulnerable to replay attacks, and we propose a practical solution.

- We test various implementations of the GAN-CAN attack in different settings on real-world driver data. Our attacks show that it is possible to fool behavior-based authentication systems with success rates up to 0.99. We also make the code of our attack and threat model publicly available at `https://anonymous.4open.science/r/GAN-CAN-1518/`.

1.4 Organization

The rest of the thesis is organized as follows. In Chapter 2, we give some background information with an overview of related works. In Chapter 3, we discuss the approaches for an authentication system available in the literature and we formalize the characteristics of the optimal target system and then we demonstrated the threat model in Chapter 4. Details on our attack approach are given in Chapter 5. We include its evaluation in Chapter 6, and in Chapter 7, we discuss possible countermeasures. Finally, Chapter 8 concludes this work and propose some possible future research ideas.

2
Background

In this Chapter, we report some background to understand the rest of the thesis better. In Chapter 2.1, we briefly overview the CAN bus technology, while in Chapter 2.2, we detail the architecture of GANs and how they can be used for malicious purposes. Behavior-based authentication systems are presented in Chapter 2.3, with a discussion on their most common characteristics. Finally, in Chapter 2.4, we conclude the Chapter with some relevant related works.

2.1 CAN Bus and ECUs

With the technological evolution of the last decades, modern vehicles employ small and simple computers called Electronic Control Units (ECUs) to command each component, from the infotainment systems to safety measures like airbags. Microcontrollers require to communicate with each other to collect information from sensors and convey instructions to actuators. Despite some effort into developing new technologies (e.g., Automotive Ethernet [28]), with rare exceptions, the connection between components happens through the CAN bus, a protocol developed by Robert Bosh GmbH in the eighties [29]. Its popularity is due to its multiplexing capabilities with only two copper cables to save on weights and its intrinsic error handling mechanisms, making the CAN bus particularly suited for safety-critical applications like vehicles. On the other hand, it natively does not provide any security features [2]. For instance, since all the messages are broadcasted, and the communication bus is unencrypted, it is straightforward

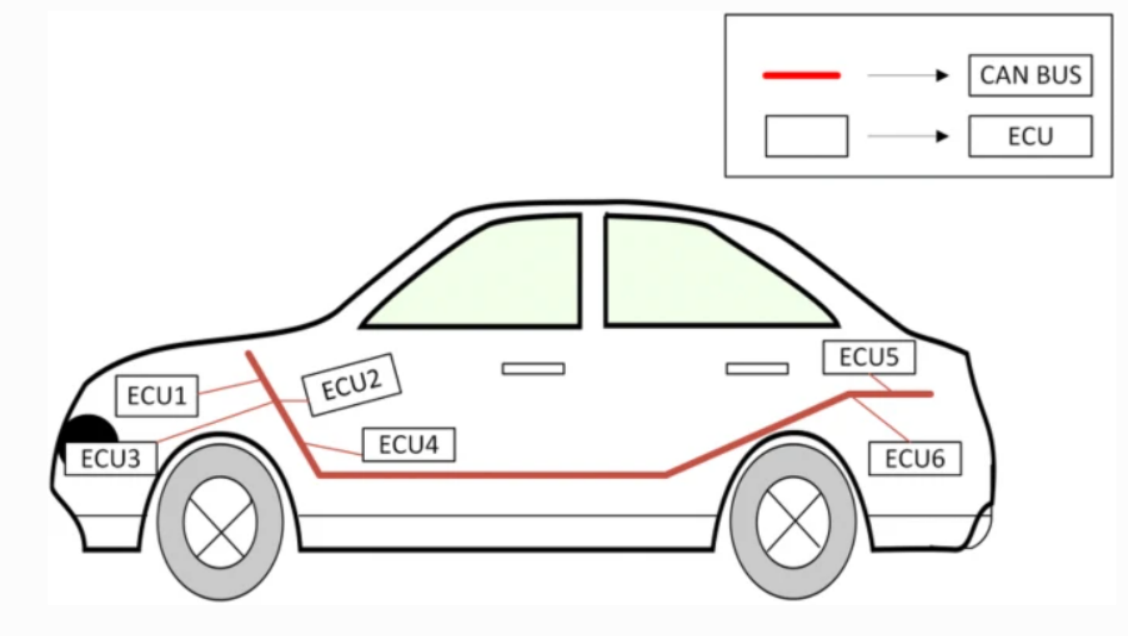
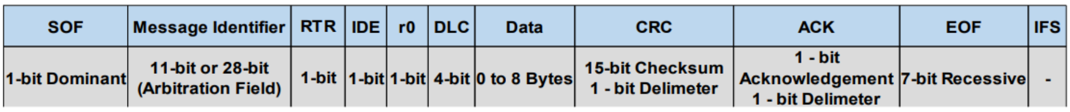


Figure 2.1: The ECUs use the CAN interface for data communication [1].

for a malicious ECU to read all the transmitted messages and inject malicious data. Figure 2.1 shows a typical scenario of a CAN bus inside a vehicle. The CAN protocol has message-based communication provided via frames, as shown in Figure 2.2. Also a complete CAN bus frame can be seen in Figure 2.3. To allow many ECU to communicate in the same bus without causing errors, the CAN bus standard [29] defines an arbitration mechanism. The transmission of each packet starts with the arbitration phase, which is won by the ECU sending the message with the lowest ID (i.e., highest priority). The other ECU will listen to the packet content and try again in the next time slot. Although the structure of the message is defined in the standard, the content of the message is designed by the manufacturer, leading to a closed environment where only the producer can interact with the bus without a big reverse engineering effort. To allow easy diagnostic of vehicles by other technicians, governments forced vehicle manufacturers to provide access to vehicle information through the Unified Diagnostic Services (UDS). It provides a standardized way to interact with every vehicle to collect information about its state and fix some anomalies by general technicians. Moreover, the UDS protocol bus is easily accessible from the OBD-II port, a connector positioned in a user-friendly location, generally below the steering wheel. Usually, this connection offers access to the CAN bus as well, even if, in modern vehicles, there could be a gateway in the middle to separate the two networks.



SOF	Message Identifier	RTR	IDE	r0	DLC	Data	CRC	ACK	EOF	IFS
1-bit Dominant	11-bit or 28-bit (Arbitration Field)	1-bit	1-bit	1-bit	4-bit	0 to 8 Bytes	15-bit Checksum 1 - bit Delimeter	1 - bit Acknowledgement 1 - bit Delimeter	7-bit Recessive	-

Figure 2.2: Classical CAN frame structure [2].

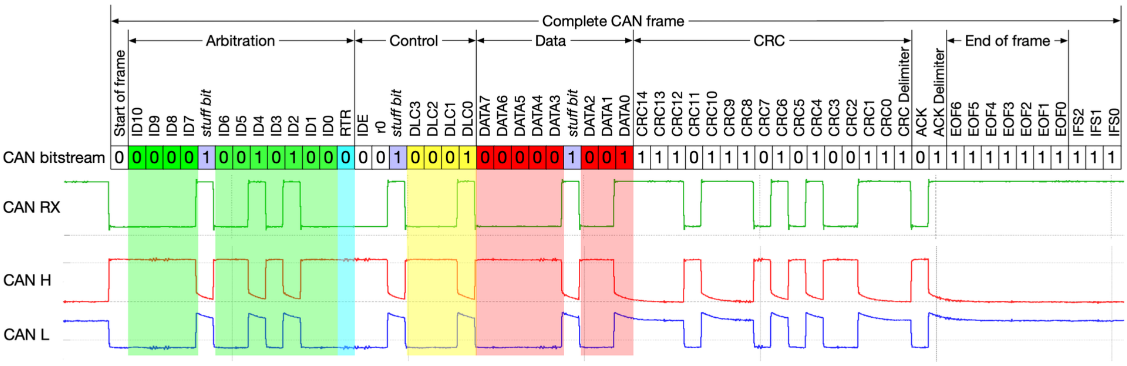


Figure 2.3: Complete CAN bus frame [3].

2.2 Generative Adversarial Networks

GANs emerged as a powerful framework for generating realistic, high-quality synthetic data. Introduced by Goodfellow et al. [30], GANs consists of two neural networks, i.e., a generator and a discriminator, engaged in an adversarial game. The generator aims to produce synthetic data samples that resemble real ones, while the discriminator is trained to distinguish between real and fake samples. A traditional GAN training process can be seen in Figure 2.4 . The generator in this context does not have direct access to real data and its learning process solely relies on its interaction with the discriminator but the discriminator has access to both the generated samples and actual samples [31]. Through an iterative training process, GANs learn to generate increasingly convincing results, which has led them to be widely adopted in various domains, including computer vision [32], natural language processing [33], and data synthesis.

However, the impressive capabilities of GANs have raised concerns about their potential malicious use. Indeed, GANs can be exploited by malicious actors to create disruptive or harmful outcomes in several domains. For instance Zhang et al. [34] assess a poisoning attack in a federated learning system using GANs. The attacker trains a GAN to mimic other participants' data, then generates poisoning updates to compromise the global model. Despite poisoning, the global model retains over 80 percent accuracy on tasks. This highlights the need for robust

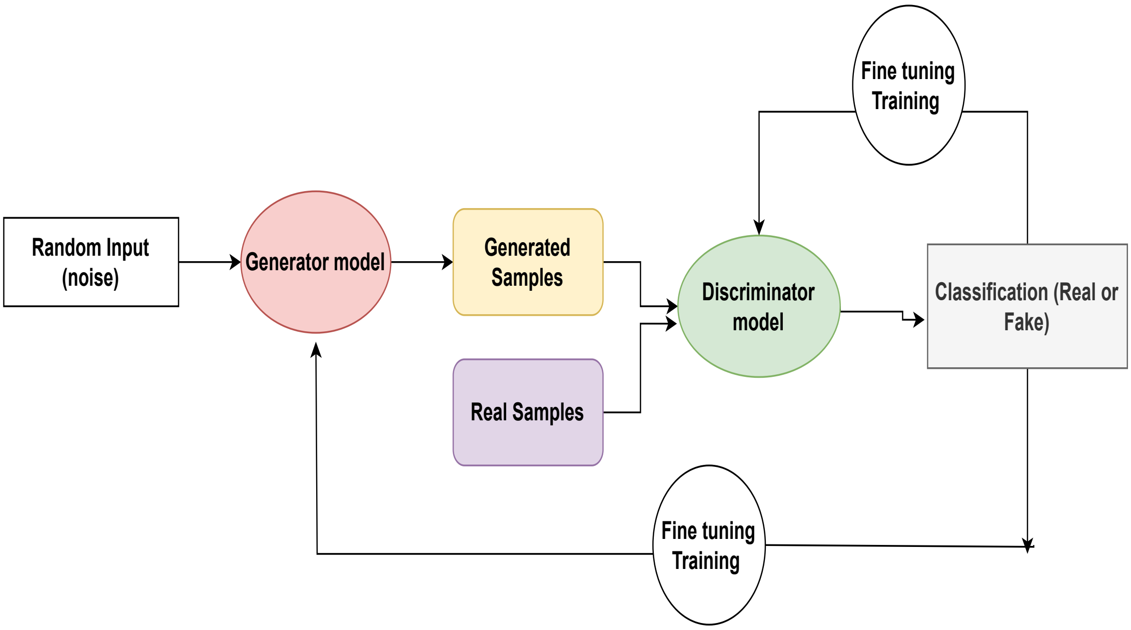


Figure 2.4: Traditional GAN Training Process.

defenses against GAN-based attacks in federated learning systems. Another major area of concern is the generation of adversarial examples for evasion attacks [24]. By exploiting GANs, attackers can generate perturbed inputs that fool machine learning models. In addition to evasion attacks, GANs can be employed to generate synthetic data that can be used to spoof or manipulate systems. It can have significant implications in applications involving authentication and identification systems.

2.3 Behavior-Based Authentication

Behavior-based authentication systems are an innovative approach to improving security in various domains. Unlike traditional authentication techniques, which rely on static credentials (e.g., passwords, tokens), behavior-based authentication methods leverage unique behavioral patterns and characteristics of individual users. This allows the system to establish the identity of each user and grant access to services or resources accordingly. It is well suited for continuous authentication and re-authentication, that is, checking the presence of the legitimate user during its usage of a device like, for instance, a smartphone [35] or a computer [36].

This kind of authentication has been tested in different domains. The most researched domain is related to computers which provide many behavior data of the user based on the keystroke dynamics [37] and mouse movements [38]. Soft keyboards are another perfect device to extract the user's behavior: the exact point pressed inside the key can be used to identify a user [39]. In general, a lot of effort has been spent on developing solutions for smartphones [35], extracting user information based on their behavior of performing certain actions on the touch screens, such as gestures and signatures [40]. They could distinguish used based on how they type using features such as velocity, device acceleration, and stroke time. Swipe movements can also continuously authenticate the user after one-time traditional authentication employing ML models [41]. Moreover, recent works applied behavior-based authentication to virtual reality [42, 43]. The automotive field has also been interested in this innovation [44], as widely discussed in Chapter 3.1.

2.4 Related Works

In recent years, researchers and experts have been actively investigating the vulnerabilities and potential threats associated with CAN bus networks. This section reviews notable research efforts in three distinct areas: CAN bus security, the deployment of Intrusion Detection Systems (IDSs) on CAN bus networks, and the emerging challenge of adversarial attacks targeting IDS mechanisms. It also presents insights into potential vulnerabilities and some of the solutions proposed in the literature.

2.4.1 CAN bus security

CAN has become a target of malicious attacks due to its critical role in vehicle control systems. To protect vehicles, conventional security mechanisms such as system lock-down, security hardening, and Intrusion Detection Systems (IDS) are often employed on ECUs. However, these measures do not effectively address attacks originating from the OBD-II port, which serves as an interface enabling direct communication with the CAN bus from external sources, including the internet [45]. Several studies highlighted the vulnerabilities and possible attack vectors on the CAN bus [2]. A malicious attacker connected to the CAN bus can force other ECUs to stop sending messages, with the so-called *bus-off* attacks [46, 47]. Such attacks can impact different components of the vehicle having catastrophic consequences. Dariz et al. [48] investigate the joint design of integrity and encryption for securing a typical CAN network with real-time traffic considerations. The study aims to address the challenges of data protection and authenticity in resource-constrained environments. Fassak et al. [49] present a secure protocol for authenticating ECUs in the CAN bus using elliptic curve cryptography (ECC). The method establishes session keys to introduce message source authentication, showing improved performance compared to existing methods in bus load. Miller and Valasek [16] demonstrated the possibility of attacking a vehicle and injecting malicious packets in the CAN bus remotely through vulnerabilities in wireless communications interfaces. Farag [50] implemented the security feature for CAN using an intuitive algorithm. This algorithm encrypts the 8-byte payload data with a dynamically changing symmetric key, synchronized across all nodes. This approach ensures that the encrypted message remains unique at any given time. In [51],the authors propose CANeleon, a novel protection scheme for defending CAN against smart attackers injecting malicious frames with legitimate frame IDs. CANeleon allows legitimate nodes to shift the spoofed frame ID, exposing malicious frames for filtering without protocol modifications. Halabi et al. [52] propose a lightweight encryption solution inspired by Blockchain technology. It generates keys based on a CAN frame's payload and the previous key, ensuring robustness. Many attack surfaces related to remote attacks in modern vehicles have been found by Checkoway et al. [53] and can lead to remote control and location tracking. For instance, Rouf et al. [54] exploit data transmitted by the tire pressure monitoring systems to track a vehicle.

2.4.2 Intrusion Detection Systems on CAN bus

To detect in advance these attacks in the CAN bus and mitigate their effects, researchers have proposed several Intrusion Detection Systems (IDSs) [55], security mechanisms designed to detect and respond to unauthorized or malicious activities. Gmiden et al. [56] discussed that ECUs connected to the CAN bus regularly send messages with unique IDs. Their simple IDS tracks the arrival time of these messages and compares the interval between each message to the expected interval and if the interval is shorter than expected, an alert is triggered. Casillo et al. [57] developed a prototype IDS system for vehicles, utilizing embedded hardware. This system employs bayesian networks, to identify and classify malicious messages transmitted on the vehicle's CAN bus network. Lampe et al. [58] developed a practical and affordable IDS for the CAN bus, in the form of an android app with easy vehicle integration. It alerts consumers of any suspicious communication, making it accessible to the average user. Jin et al. [59] introduce a light-weight IDS that relies on signatures and can be seamlessly and promptly implemented on the ECUs of vehicles. Islam et al. [60] propose a novel graph-based Gaussian naive Bayes (GGNB) intrusion detection algorithm that utilizes graph properties and PageRank-related features for attacks against CAN bus. Caivano et al. [61] analyze an optimized and efficient network-based IDS for detecting CAN bus attacks using quantum annealing. The quantum annealing algorithm outperforms classical classification algorithms in terms of time performance, which is crucial for identifying attacks in the automotive sector. In [62] researchers present an algorithm which measures inter-packet timing over a sliding window and compares average times to historical averages, generating an anomaly signal. Zhao et al. [63] present a series of methods using two classifiers trained with Auxiliary Classifier Generative Adversarial Network (ACGAN). These methods aim to detect and assign fine-grained labels to known attacks, as well as identify the unknown attack class. Longari et al. [64] propose an IDS based on LSTM autoencoders to detect anomalies and possible attacks by comparing reconstructed data streams with the real ones. Another LSTM approach is given by Hossain et al. [65], which propose an IDS able to detect several types of attacks (e.g., denial of service, spoofing, or malfunctions).

2.4.3 Adversarial attacks against IDS

Given the widely adopted usage of ML or DL models for IDSs, these systems can be vulnerable to adversarial attacks, which are a category of well-researched attacks in the literature [66]. To address this issue, Pawlicki et al. [67] propose a methodology to detect several adversarial attacks in different IDSs. Instead, several research works focus on using GANs to fool the detection systems. For instance, polymorphic Distributed Denial of Service (DDoS) attacks can be carried out using those generative networks, as shown by Chauhan et al. [68]. Shu et al. [69] present the Gen-AAL algorithm, which evaluates the susceptibility of ML-based IDS systems to adversarial attacks which unlike other methods, it can utilizes active learning and GANs to generate adversarial examples without requiring prior knowledge of the IDS model's internal structure or loss function. Apruzzese et al. [70] examine machine learning-based network IDS to understand the real capabilities and conditions required for successful adversarial attacks. By evaluating existing literature, it highlights the limitations and strengths of proposed adversarial attacks. Lin et al. [71] propose IDSGAN, a generative adversarial network framework for generating adversarial malicious traffic records to deceive and evade IDSs. IDSGAN performs black-box attacks without knowledge of the detection system's internal structure. Zhou et al. [72] present a novel Hierarchical Adversarial Attack (HAA) method for Graph Neural Network (GNN)-based intrusion detection in IoT systems with limited budgets. The approach involves a shadow GNN model, a saliency map technique for generating minimal perturbations, and a hierarchical node selection algorithm based on random walk with restart (RWR) to target vulnerable nodes. In [73] the authors propose a novel framework named FGMD that offers defense against adversarial attacks by utilizing feature grouping and multi-model fusion techniques. Mohammadian et al. [74] present a new approach for conducting adversarial attacks against deep learning-based malicious network activity classification. The authors utilize the jacobian saliency map to identify the best group of features, with varying features and perturbation magnitudes, to generate effective adversarial examples. Debicha et al. [75] examine the actual feasibility of evasion attacks, specifically targeting Network-based Intrusion Detection Systems (NIDS). The researchers sought to demonstrate that their proposed adversarial algorithm could effectively deceive these machine learning-based IDSs. The evaluation was conducted in a black-box setting, where various constraints were considered to showcase the practicality of such attacks. More generally, it is demonstrated by He et al. [76] that IDSs can be fooled by artificially generated data even with restricted adversarial capabilities. Also Usama et al. [77] propose an adversarial machine learning attack utilizing GANs to evade a machine learning-based IDS successfully. Furthermore, they demonstrate that GANs can be employed to fortify the IDS against adversarial perturbations.

3

Behavior-Based Driver Identification System

In this Chapter, first we delve deeper into the behavior-based driver identification systems proposed in the literature. In Chapter 3.1, we take a closer look at the characteristics and security aspects of these systems, viewing them as practical anti-theft measures in the real world. Then in Chapter 3.2.1, we present the authentication system's pipeline, outlining its main components. Subsequently, in Chapter 3.2.2, we explore two system model scenarios to better understand the system's behavior and vulnerabilities. Finally, in Chapter 3.2.3, we provide detailed insights into the implementation of the authentication device, emphasizing its essential features and functionalities.

3.1 Toward a Perfect Behavior-Based Driver Identification System

A vast amount of literature considers the problem of identifying the driver behind the wheel based on behavioral data. This could be useful in identifying and blocking vehicle thefts but can also be used to offer more personalized experiences to drivers. For example, it can be used to set some parameters of the vehicles based on user preferences [78] or to collect data to optimize and enforce driver-based insurance calculating risk factors on the fly [17]. However, except for some of the works [79, 17, 80, 81], the main target is identifying the driver.

Most approaches extract CAN bus data via the OBD-II port which can provide a wide va-

Goal	Type	Model	References
Style	Clustering	Clustering	[81]
	Statistical	Time-domain	[80]
	DL	CNN, LSTM	[17]
Driver	DL	LSTM	[86, 15, 87, 78]
		RNN	[78, 88, 89, 23]
		CNN	[90, 89]
		SVDD	[90]
		AdaBoost	[91]
		Autoencoder	[92]
		DeepRCN	[14]
		4-layer MLP	[93]
		GAN	[94, 95]
	ML	GMM	[96]
		k-means	[97]
		KNN	[82, 22, 98, 99, 84]
		RF	[83, 98, 99, 84]
		SVM	[98, 99, 84, 100, 21, 20]

Table 3.1: Summary on the different models employed by papers in the literature. The paper's target can be to profile the driving *style* of the driver or to identify the *driver*.

riety of sensors and actuators readings to characterize the vehicle state. Many researchers do not provide access to the dataset they used, making it complicated for the community to reproduce and improve the results, other works employ publicly available datasets. The most used one comes from the OCSLab [4, 82] and comprises 54 sensors reading from the vehicle bus, including ten different drivers. Other less considered datasets contain different data types, such as stability [83] and Global Positioning System (GPS) data [84]. Few works also employ physiological data [85], even if its applicability in a real scenario is challenging. In our work, we will use the OCSLab [4] dataset to make our results easily comparable with others in the literature.

Almost all the works attempting to identify the driver employ ML or DL algorithms, as summarized in Table 3.1. The paper by Erzin et al. [79] is one of the first to discuss how features such as pressure on pedals, vehicle speed, engine speed, and steering angle can be combined to identify a driver. They state that authentication should be done before the vehicle moves. However, these data can be solely employed to verify the driver's state (sleepy, active, etc.). In [83], the authorized driver has a specifically-trained ML model containing their profile information. They do binary classification and show that not only they can authenticate drivers, but they can also extract features such as gender from the trained model. The authors of [100] use CAN bus data of an Electric Vehicle (EV), particularly focusing on pedal operation patterns and GPS traces of different drivers driving on the same route. The authors use such data to implement an Support Vector Machine (SVM) and Universal Background Model (UBM)-based ML model to authenticate users. In [90], Xun et al. propose driver fingerprinting to authenticate a user in real-time using CAN bus data. The authors use DL methods such as Convolutional Neural Network (CNN) combined with Support Vector Domain Description (SVDD) to detect illegal drivers. Other ML models have employed with discrete success, such as k-Nearest Neighbors (kNN) [82, 22, 98, 99, 84], Random Forest (RF) [83, 98, 99, 84], and Gaussian Mixture Model (GMM) [96]. Better successes have been observed with DL models, especially employing LSTM [86, 15, 87, 78] or RNN [78, 88, 89, 23]. Other models have been tried as well, such as AdaBoost [91], Autoencoder [92], and GANs [94, 95]. In [95] researchers introduce a new GAN model, Convolutional Long short-term GAN (CLGAN), which combines LSTM and CNN. This model excels in feature extraction and preservation while minimizing overfitting, showing promise for generating high-quality outputs.

Together with the models, the number of features to be employed has also been analyzed in the literature. Marchegiani et al. [100] conduct some tests to assess the feasibility of identifying a driver with one feature only (e.g., acceleration or brake) using SVM. Rahim et al. [84] employ GPS data only, training ML models with solely three features (orientation change, stable speed, total acceleration), but still reaching up to 90% accuracy in detecting drivers. Thanks to its complexity, DL are usually more suited to manage a higher number of features. For example, Chen et al. [92] employed 51 features available in the OSCLab dataset to train an Autoencoder, while Ravi et al. [15] employs 40 features in a LSTM.

Several preprocessing techniques have been employed to make the best use of the data. Miyajima et al. [96] employ cepstral spectral features, generally used for speech and speaker recognition, associated with a GMM model. Fugiglando et al. [80] employs a time-domain analysis of the features to extract the maximum entropy to characterize the drivers. DL and LSTM usually require less preprocessing effort since the model can automatically capture the peculiar feature of the data. Most works in the literature divide the data with a sliding window approach to be better analyzed by the models. The window size may vary from 12 seconds [91] to 300 [90]. Moreover, to enhance the number of samples, some papers maintain an overlap between adjacent windows [14, 97, 82].

Another factor impacting these systems is their implementation in real-world scenarios. Most of these works do not specify which system component is responsible for driver authentication.

Although simple ML algorithms can be executed on cars, DL networks based on complex and deep structures might require prohibitive costs for the execution on a vehicle and are more likely to be implemented on a dedicated and resourceful server, at least for the training part. El Mekki et al. [78] implemented a proof-of-concept authenticator with Automotive Grade Linux [101], showing its easy applicability to UI personalization. However, no one ever implemented or discussed in detail how a physical anti-theft device based on driver behavior can be developed.

Since running and, especially, training ML and DL models can be expensive, some literature works delegated at least one of these tasks to a central server in the cloud. For instance, Kwak et al. [82] propose to implement the anti-theft module in a remote server accessible via the internet. The car sends via wireless communication driver behavior CAN bus data to the server, which then extracts specific features and feeds them to a previously trained ML model. Based on the same system model, Ezzini et al. [99] proposed a driver authentication scheme implemented on an internet-connected dedicated server that extracts predefined features from CAN bus data to authenticate the driver. In other works [78, 86, 102, 97], a remote server has a part in the authentication process. All these works, however, do not mention security on the channel between the vehicle and the server, ignoring the threat imposed by sharing users' sensitive plaintext data over a wireless channel which may expose them to possible thefts by malicious users.

3.2 Authenticator System Model

We now discuss the characteristics that an anti-theft system should have to be secure as a real-world device. First, we show the pipeline of the authentication system in Chapter 3.2.1, while in Chapter 3.2.2, we discuss two possible deployment scenarios. Then, in Chapter 3.2.3, we provide more details on the possible implementation by defining the used models and their parameters.

3.2.1 Pipeline

In Figure 3.1, we show the pipeline of the behavior-based anti-theft system. The first step is the collection of the raw data from the CAN bus, which can be performed in different ways, e.g., from OBD or directly from the CAN bus. The second stage envisions data processing, i.e., reshaping the collected data into suitable data structures for the classification model. The third

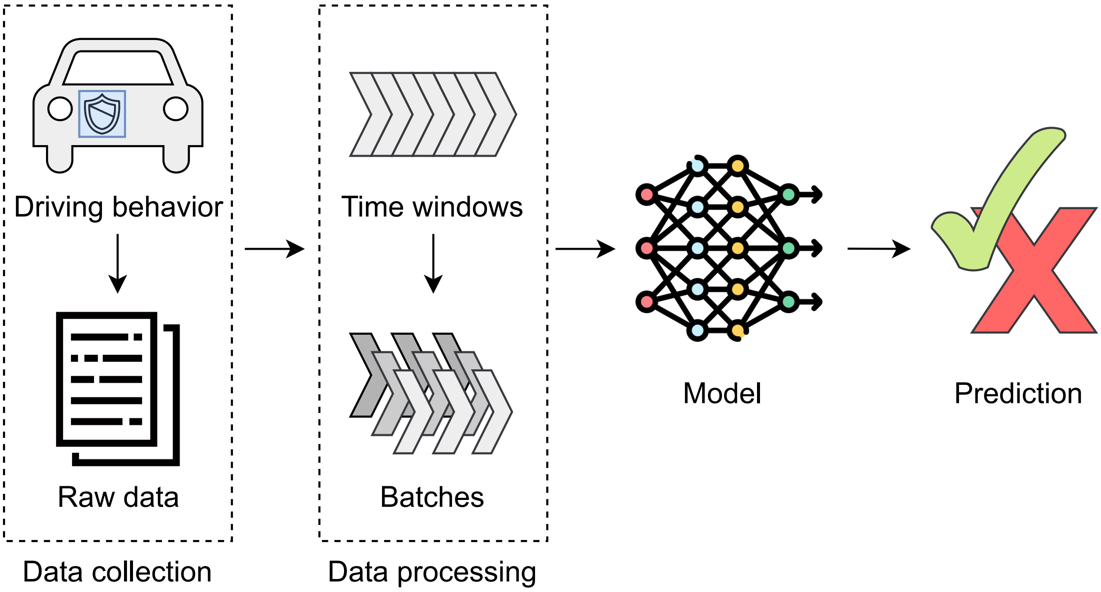


Figure 3.1: Pipeline of the authenticator system.

phase is model training. As typical in the literature [14, 20], we assume that the data collection phase is secure, i.e., no training data poisoning. After training, the model is ready to be tested and, based on the implemented model and collected data, may be able to recognize more than one driver.

Input data must be processed to extract the features that the authentication model will use. In Chapter 3.1, we show many works in the literature that use LSTM models, given their ability to grasp causality between data samples. For this reason, sequential data must be aggregated in different time windows, which will then be fed to the classifier in batches. Here, the model will process each batch to generate a prediction, i.e., determine whether the driving data comes from a legitimate driver.

3.2.2 System Model Scenarios

The overall system model we analyze in this thesis comprises a behavior-based authenticator system, employed as an anti-theft mechanism, that continuously authenticates the driver against a list of known and acceptable drivers. As depicted in Figure 3.2, the authentication system can be deployed inside the vehicle's network in the two following configurations.

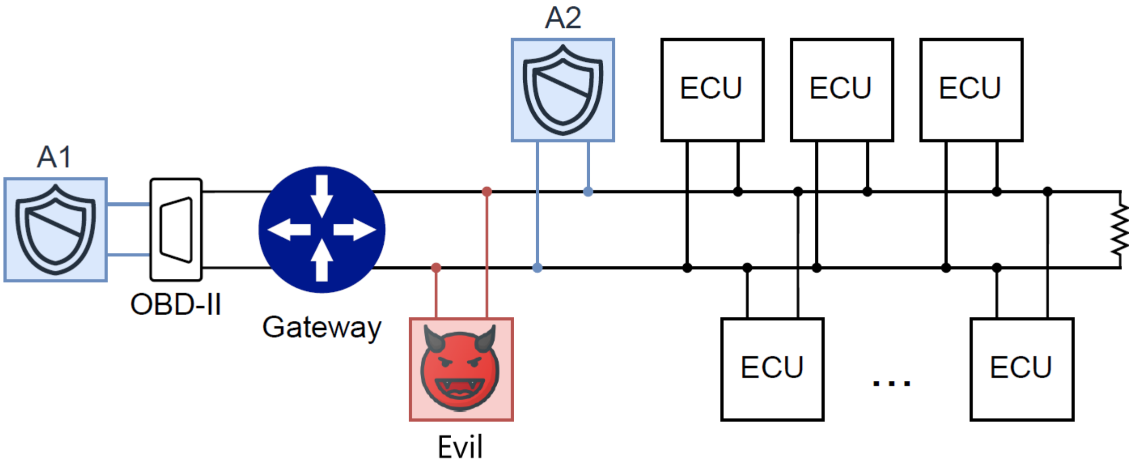


Figure 3.2: A schema of the system and threat model.

External Authenticator (A1). The easiest way to connect the anti-theft module is via the OBD-II port. Thanks to UDS capabilities, such a system is interoperable between different vehicles because of its standardized data structure, which is employed in all vehicles in the market. Ideally, a vehicle owner can buy the anti-theft device even after the vehicle's purchase and easily configure it. The ease of configuration further simplifies the setup process, enabling owners to install and activate the anti-theft module without the need for specialized knowledge or complex installations. Such a solution can run on a cheap microcomputer (e.g., a Raspebbery Pi [103]), employing a cloud computing service for the one-time training during the configuration phase. To prevent an attacker from simply disconnecting it from the OBD-II port, the anti-theft must be equipped with anti-tampering and physical intrusion detection solutions, such as [104]. However, developing such technology in an almost plug-and-play device could be difficult. This could expose the device to Man-in-the-Middle (MitM) attacks through which an attacker can freely tamper with all the UDS responses, launching simple replay attacks to break the authenticator.

Internal Authenticator (A2). The connection of the authenticator to the internal network is inherently more robust. Removing or disabling a hidden device requires a higher level of expertise and effort, making it a more daunting task for attackers to accomplish. Moreover, anti-tampering techniques developed by the manufacturing company during the vehicle design can be more effective compared to those developed by a third party [105, 104]. Being connected to the main CAN bus, the authenticator is exposed to the raw packets sent by ECUs to control vehicle parameters and regulate driving. An attacker aiming at tricking it is then forced to send packets to the main bus and deal with its consequences. In fact, a simple replay attack cannot be feasible without compromising driver safety, as better explained in Chapter 5.1. Conversely, it is worth mentioning that such a solution can be implemented only by the manufacturing company since the data format is usually proprietary and not publicly available. Even if there has been some effort in reversing packet formats of cars [106], the physical deployment inside the vehicle could be tricky. For sure, it will be less convenient than a plug-and-play solution like A1.

Independently of its location, the authentication system can collect generic data commonly used in the literature for behavior-based driver authentication without restricting to a specific subset, e.g., pedal pressure, wheel steering, engine Revolutions Per Minute (RPM). We consider the general framework, where every data exchanged on the CAN bus or requestable through the UDS protocol can be collected and suitably combined for authentication purposes. In particular, the authentication model is periodically fed with a series of values representing the vehicle *state*. A *state* comprises the most recent value for each feature collected from the data.

Depending on the capabilities of the device installed in the vehicle, authentication can be done locally inside the device or delegated to a third party. In the latter case, data can be reported to a central authentication server through an internet-based wireless connection which will feed the cloud model with the driver's behavioral data and reply with a decision. In the second case, as typical in the literature using this system model, the communication between the vehicle and the authentication server is not secured [82, 94]. However, differently from the existing literature, we strengthen the system's security by considering an encrypted connection to a central server that can recognize the sender and infer its legitimacy.

3.2.3 Authentication Device Implementation

Grasping the best insight from the many works on behavior-based driver identification in literature (Chapter 3.1), we develop our solution for a robust authentication system.

Training Data. In order to ensure consistency with previous research works, we leverage a frequently employed dataset called OCSLab [82]. This dataset comprises 94380 data points, thus accounting for around 26 hours of driving. Initially, we considered a dataset with 54 features, but after careful analysis, we discovered that eight features had either zero or constant values throughout the dataset. Realizing the negligible variation and significance of these eight features, we concluded that they would not contribute meaningfully to the authentication process. Consequently, we made the decision to filter out these eight features resulting in a final set of 46 features for the baseline authentication. In addition to feature selection, we perform windowing on the time series data. This process involves dividing the dataset into windows of a specified size. The windows are created by iterating through the data with a step size of `window-size/2`. We consider a window size of 16 seconds, which is slightly smaller than the average found in the literature but it offers distinct advantages for our classification purposes. The choice of a smaller window size enables us to perform the classification more frequently, allowing us to gain insights into driving behavior patterns in finer-grained intervals.

The data is organized into batches, each containing four time windows. Each window represents driving behavior observations collected over 16 seconds. To ensure data quality and the suitability of the dataset for the authentication models, essential pre-processing steps, such as normalization, are applied to the dataset to ensure data quality and suitability for the authentication models. By doing so, we eliminate any potential bias that may arise due to differences in the magnitude of features, ensuring that all features contribute equally to the authentication models.

Models. We re-implemented several authentication models proposed in the literature to establish the best system. We selected the top three performers from these models based on their reported accuracy and effectiveness [78, 23, 88]. By selecting the top performers, we aimed to ensure that the final authentication system we adopt is robust, reliable, and capable of delivering accurate results in identifying authorized users based on their driving behavior. The chosen models for the baseline authentication are Long Short-Term Memory Fully Convolutional Network (LSTM-FCN) [78], LSTM [23], and RNN with Gated Recurrent Unit (GRU) [88]. After pre-processing the data by aggregating it in time windows and batches, we train and test the selected models on the dataset. We employed an ensemble approach to further enhance the robustness and reliability of the authentication system. We combined the outputs of the three selected models using majority voting, and the ensemble performed better than all three models. Moreover, using the ensemble of models with majority voting allows us to create a black-box scenario for the attack. In a black-box setting, the attacker has limited knowledge about the inner workings of the authentication models. By generating data that can defeat each of these individual models independently, we demonstrate the effectiveness and versatility of our attack approach. This demonstrates the potential vulnerabilities and limitations of the baseline system, thereby highlighting the importance of further enhancing the authentication mechanisms to mitigate adversarial attacks.

Implementation. As stated in Chapter 3.2.2, the most secure implementation of the authentication device envisions its connection inside the CAN bus (i.e., A2). Thus, after collecting data from the legitimate driver behavior to perform training (which can require driving for several kilometers), the system can finally be deployed for authentication. In the OCSLab dataset, the driving data has been collected during a round-trip of about 46km (around 2.5h of driving) for each driver. However, it might be possible to shorten this length by considering a round-trip that includes different driving behaviors (e.g., highway driving, urban roads, traffic conditions). As done in the OCSLab dataset, a single data sample (i.e., a value for each of the 46 features) is extracted each second. Considering the window size of 16 seconds, window advancement with 8 seconds step size (i.e., 8 seconds overlap with the preceding window), and the batch size of 4, it leads the classification model to take a decision every 40 seconds of driving. This time interval allows the system to evaluate and classify driving behavior at regular intervals, enhancing the responsiveness and real-time nature of the authentication process.

Once a non-authorized batch of data is detected, it triggers appropriate actions to ensure the security of the vehicle and its occupants. One potential action is notifying the vehicle owner immediately, alerting them to the unauthorized access attempt. Alternatively, the system may decide to initiate a more assertive response, such as pulling over the vehicle and blocking further driving attempts. However, several aspects must be considered when dealing with possibly unidentified behaviors.

Model Errors – The usage of DL models for authentication leads to high accuracy scores, as discussed in the literature. These results, however, are rarely perfect (i.e., attaining accuracy 1) and they are not immune to misclassifications. Despite their high accuracy, there will always be a percentage of wrong predictions or misclassifications, especially when dealing with challenging and ambiguous driving scenarios. For usability and security reasons, both the rate of false positives (i.e., legitimate driver classified as not authorized) and false negatives (i.e., illegitimate driver classified as authorized) should be kept at a minimum. Thus, depending on the baseline evaluation of the authentication system, notifications might be sent, or countermeasures might be taken after detecting two or more non-authorized batches. This decision threshold helps ensure that the detection is more robust and accurate, reducing the chances of acting on isolated or occasional misclassifications. Implementing a threshold for the number of non-authorized batches helps strike a balance between maintaining high security standards and avoiding unnecessary interruptions for legitimate vehicle owners. It allows the system to discern genuine security threats from occasional anomalies in driving behavior that may occur due to external factors or benign variations.

Early Driving Stages – The driving data used to train the models should be collected once the vehicle is moving on the road. Indeed, early driving stages (e.g., idle state, coming out of a parking lot) often do not contain any relevant data to identify the driver and thus might lead to errors and misclassification. During the early driving stages, the vehicle may exhibit minimal driving behavior, and the data collected during these periods might lack distinctive patterns or features that are crucial for driver identification. For these reasons, the authentication system should ignore predictions performed when the vehicle is not moving for extended periods or when not on the road.

4
Threat Model

Based on the knowledge that the attacker has on the victim profile and the authentication system, we can define four different cases, as summarized in Table 4.1.

4.1 White Box (WB): Known Model and Data

In the White Box (WB) scenario, the attacker knows the architecture of the authentication system and the data generated by the victim driver. The attacker can obtain the architecture from either: i) known implementation disclosed by the manufacturer, ii) having direct access to the hardware memory in case no Trusted Platform Module (TPM) is used, iii) having direct access to the authentication system input/output and using model extraction techniques (from local [107, 108] or from cloud [109]). Regarding data, the attacker can obtain data fed to the local authentication system by either monitoring the data exchanged over the CAN bus (e.g., mounting a malicious ECU or compromising one already installed [16]) or by eavesdropping

	Model	**Data**	**Difficulty**
White Box (WB)	✓	✓	Easy
Gray Box 1 (GB1)	✗	✓	Medium
Gray Box 2 (GB2)	✓	✗	Medium
Black Box (BB)	✗	✗	Hard

Table 4.1: Different cases based on the attacker's knowledge.

communications between the car and the authentication server, if not encrypted.

4.2 Gray Box 1 (GB1): Unknown Model, Known Data

In the first Gray Box (GB1) scenario, the attacker can obtain data fed to the authentication system by legitimate drivers but cannot obtain in any way the architecture behind the authentication system. An example of such a scenario is a cloud authentication system getting data through an unencrypted wireless channel from a vehicle [82, 94]. In this context, legitimate drivers interact with the cloud authentication system through the wireless channel. However, the data transmitted between the vehicle and the cloud is susceptible to eavesdropping by malicious actors. An attacker with the capability to intercept and monitor the communication can obtain sensitive information that is being sent from the vehicle to the cloud for authentication.

Despite the attacker's ability to intercept this data, the inner workings and architectural details of the authentication system itself remain hidden from their reach. This means that the attacker cannot directly access or retrieve the model responsible for performing the authentication within the cloud infrastructure. The authentication model remains secure and obscured from external entities, preventing unauthorized access to its sensitive components and logic.

4.3 Gray Box 2 (GB2): Known Model, Unknown Data

In the second Gray Box (GB2) scenario, the security landscape shifts, presenting a different set of challenges for the authentication system. In this case, the attacker gains the ability to access the architecture of the authentication system itself. This means that the internal structure, algorithms, and design of the system become exposed to the malicious user, possibly through the exploitation of techniques similar to those described in the White Box (WB) case. Despite obtaining knowledge of the authentication system's architecture, the attacker faces a limitation – they are unable to directly access any data from the original driver. This means that they do not have access to the input data that legitimate drivers use during the authentication process. To enable such an attack, the malicious user can leverage model extraction attacks to get the target ML model [109, 110].

	A1	A2
WB	Replay	Smart-Replay
GB1	Replay	Smart-Replay
GB2	GAN-CAN, and then replay	GAN-CAN
BB	GAN-CAN, and then replay	GAN-CAN

Table 4.2: Which is the easiest attack that will work in a certain scenario? With *GAN-CAN, and then replay*, GAN-CAN is only needed to generate one sample to be submitted as a replay attack to the authenticator.

4.4 Black Box (BB): Unknown Model and Data

The Black Box (BB) case is the trickiest one. The attacker cannot freely access the authentication system's or the legitimate driver's data. However, the malicious entity can read responses from the authentication system as in all the other cases.

In our threat model, we consider a malicious attacker aiming to get authenticated even if not in the designated driver's list. Table 4.2 summarizes the easiest attack for each location of deployment of the authenticator and each attacker scenario. The A1 case is the easiest since a simple replay attack is always sufficient to trick the model. However, in GB2 and BB cases, the attacker has no legitimate driver data. Therefore, they must employ GAN-CAN to obtain a valid state sample to be replayed to the authenticator. Note that, in the A1 case, the attacker can send whatever data they need to the authenticator without caring about side effects since the malicious data would never reach the internal bus. This is different in the A2 scenario, where the attacker is limited to modifying only certain features not to compromise driving safety while stealing the car. In those cases, an attacker cannot just replay the whole legitimate traffic but always needs a more sophisticated solution. Still, in WB and GB1 scenarios, a *smart*-replay attack is possible, in which only specific features are modified to perform the attack. Due to A2 scenario's higher complexity, we will present only results related to this attacks.

Finally, we assume the attacker has access to the internal CAN bus of the victim vehicle and know the data format of the vehicle's bus [106]. This is a fair assumption since attackers can physically access the CAN bus in various internal and external vehicle points, such as headlights [6]. The access can also be obtained remotely by compromising a connected ECU, for instance, exploiting modern vehicles' high connectivity [16]. These unauthorized vehicle accesses underscore the need for behavior-based authentication systems and reveal the shortcomings of traditional defense methods. In all cases, the attacker will get complete access to the network since communications inside the CAN bus are neither authenticated nor encrypted [29].

5
GAN-CAN Methodology

We now discuss the implementation of the **GAN-CAN** attack. While not adhering to the traditional GAN training framework, our generator models are optimized to bypass a fixed discriminator (i.e., the authentication system) rather than incrementally training both components. Thus, the deployed generative networks that are detailed in this section leverage the discriminator feedback (i.e., the authentication system output) as a means of optimization and loss computation. Chapter 5.1 discusses feature safety, which is essential to understanding attack implementations. Based on the scenarios presented in Chapter 4, we divide the specification of our methodology into three scenarios: GB1 in Chapter 5.2, GB2 in Chapter 5.3, and BB in Chapter 5.4. The attacks are summarized in Figure 5.1.

5.1 Features Safety

As part of our study, we encounter a crucial safety concern about data injection into the CAN bus. Because each packet sent alters the vehicle's driving state, we need to avoid injecting packets that may impact the vehicle's operation and the driver's safety during the attack. Therefore, we consciously select a subset of features enabling us to inject our malicious data into the CAN bus while mitigating potential safety risks. For this reason, we divide the features into three main categories: *modifiable*, *borderline*, and *non-modifiable*, as depicted in Table 5.1. The *modifiable* category indicates that features can be altered or manipulated without affecting the driver experience. These features offer total flexibility in terms of modifying their values. The modifiable features are carefully selected and engineered to have minimal influence on the overall driving behavior or vehicle performance. Conversely, features classified as *non-modifiable* cannot be directly modified due to their inherent nature or external constraints. Tampering with these features can change the vehicle's state in ways that jeopardize the driver's safety. Non-modifiable features are critical parameters that directly influence the vehicle's performance, safety, and overall functionality. Finally, the *borderline* category includes features that lie in a gray area between *modifiable* and *non-modifiable*. These features have certain constraints or limitations on their modification. For example, they may be partially modifiable within certain predefined ranges or subject to specific conditions.

When conducting our attack, we do not want to compromise the driver's safety. Therefore, we focus solely on the 22 *modifiable* features out of 46 (i.e., the 48%). We have consciously decided to exclude features categorized as borderline and not modifiable due to safety concerns. By prioritizing safety, we acknowledge the potential risks of modifying certain features, particularly those related to critical vehicle systems or external environmental factors. Nonetheless, we train the authentication model using all 46 features. The intention is to establish the performance of the authentication system using a comprehensive set of features. However, when conducting the attack phase, we constrain our manipulation efforts only to the 22 identified safe features. This means that during the attack, we will exclusively focus on modifying and injecting values into these specific features while keeping the values of the remaining features unchanged. This constrained approach adds an extra level of defense against adversarial attempts. Indeed, once the critical features have been identified (i.e., *non-modifiable* and *borderline* features), one possible countermeasure to strengthen the authentication system could be using only those kinds of features for classification. We attempted to train the authenticator models using solely the non-modifiable and borderline features. However, this approach led to a decline in accuracy on the test set, with the performance dropping to 61.7% , due to the high importance of modifiable features [15, 14]. Thus, we conclude that the *modifiable* features are still essential for efficiently implementing the authenticator.

5.2 Gray Box 1 Scenario

In this scenario, summarized in Figure 5.1a, the attacker can access the authenticated data (i.e., legitimate driving behavior) without knowing the authenticator model implementation. However, since we are working in the A2 implementation of the authenticator model, as stated in Chapter 4, the attacker cannot just replay the legitimate traffic they have access to. Thus, a

Importance	#	Features
Modifiable	22	Fuel consumption
		Short Term Fuel Trim Bank1
		Intake air pressure
		Engine soacking time
		Long Term Fuel Trim Bank1
		Engine torque after correction
		Torque of friction
		Flywheel torque (after torque interventions)
		Current spark timing
		Engine coolant temperature
		Engine Idel Target Speed
		Engine torque
		Calculated LOAD value
		Minimum indicated engine torque
		Maximum indicated engine torque
		Standard Torque Ratio
		Requested spark retard angle from TCU
		TCU requests engine torque limit (ETL)
		Target engine speed used in lock-up module
		Activation of Air compressor
		Engine coolant temperature.1
		Calculated road gradient
Borderline	15	Accelerator Pedal value
		Absolute throttle position
		Engine in fuel cut off
		Engine speed
		Flywheel torque
		TCU requested engine RPM increase
		Torque converter speed
		Wheel velocity front left-hand
		Wheel velocity rear right-hand
		Wheel velocity front right-hand
		Wheel velocity rear left-hand
		Torque converter turbine speed - Unfiltered
		Clutch operation acknowledge
		Acceleration speed - Lateral
		Steering wheel angle
Not-modifiable	9	Throttle position signal
		Current gear
		Converter clutch
		Gear Selection
		Vehicle speed
		Acceleration speed-Longitudinal
		Master cylinder pressure
		Steering wheel speed

Table 5.1: Distribution of features into safety classes.

smart-replay attack must be deployed. With this attack, the attacker can replay the legitimate traffic using only the features labeled as *modifiable* in Chapter 5.1. Instead, the data for all the other features (i.e., *borderline* and *non-modifiable*) are generated by the attacker themselves while driving the stolen car. In this way, the attacker can drive the car without any interference in their driving safety and still be authenticated by the anti-theft system.

5.3 Gray Box 2 Scenario

In this scenario, summarized in Figure 5.1b, the attacker can access the authenticator model implementation without knowing the legitimate driving behavior. Thus, the attacker can preemptively use the legitimate model to train the generator of a GAN. Indeed, it is possible to use the authenticator model as an oracle and train a neural network with its feedback, creating a model that generates legitimate fake packets from random data. This approach diverges from the conventional GAN training process. In this scenario, we do not employ a discriminator that learns in parallel with the generator. Instead, we utilize a pre-trained surrogate model that furnishes feedback to the generator. Consequently, we can categorize this as a modified GAN training process, predicated on the constraint of not having direct access to the data. The Generator's architecture consists of multiple layers as shown in Table 5.2 , each responsible for specific transformations of the input data. The network takes three main parameters as inputs: batch size, window size, and number of features. The batch size represents the number of data samples processed in each training iteration, while the window size defines the duration of each time window for driving observations. The number of features signifies the number of input features representing different aspects of driving behavior. The neural network architecture is designed to efficiently generate realistic driving behavior samples. The first layer, which consists of 128 neurons and performs a linear transformation on the input data. The LeakyReLU activation function with a slope of 0.2 is then applied to introduce non-linearity and better handle vanishing gradients during training. The second layer, consists of 256 neurons and further processes the data using another linear transformation followed by the LeakyReLU activation function. This process continues with the third layer, containing 512 neurons, and the fourth layer, which has batch size * window size neurons. The role of fourth layer is particularly significant, as it serves as the bottleneck layer, reducing the dimensionality of the data to fit within the desired window size. The network's ability to compress and generate meaningful representations within the confined window size is crucial for generating realistic and coherent driving behavior. The final layer, with number of features neurons, performs the final linear transformation, mapping the reduced data back to the original number of features dimensions. This output represents the generated deceptive driving behavior, closely resembling authentic driving patterns.

Given the vast search space and potential challenges in convergence and local minima encountered, we employed a modified reinforcement learning approach to optimize the learning procedure for our generator. Doing so enables the generator to explore the search space more efficiently and adapt its strategy based on feedback and rewards received during the learning process. This approach allows us to overcome the limitations of traditional optimization techniques and navigate the complex landscape more effectively. By employing exploration-exploitation strategies, the generator can balance between trying new approaches and exploiting existing knowledge to find promising regions of the search space.

Layer	Input Shape	Number of Neurons	Activation Function
Layer 1	num_features	128	Leaky ReLU (0.2)
Layer 2	128	256	Leaky ReLU (0.2)
Layer 3	256	512	Leaky ReLU (0.2)
Layer 4	512	batch_size*window_size	Leaky ReLU (0.2)
Layer 5	batch_size*window_size	num_features	Linear

Table 5.2: Architecture of the Generator.

The reinforcement learning parameters include the maximum episode length, the number of episodes, the learning rate α, and the discount factor γ. These parameters govern how the generator's latent input will be updated using reinforcement learning. The training loop consists of episodes, where each episode starts with initializing the latent input and the episode reward. Within each episode, the generator generates a sample based on the current latent input. This generated sample is then evaluated by the surrogate model (i.e., the authentication system). The surrogate model's output makes predictions, and random target labels are created for comparison. The reward is calculated as the mean accuracy of the predictions matching the targets. The latent input is updated using reinforcement learning by computing the temporal difference error (*td_error*) as the difference between the reward and the cumulative episode reward. The reward is a measure of how well the agent performed in a specific episode, providing immediate feedback on the quality of its decision. The cumulative episode reward, on the other hand, represents the total reward accumulated throughout an entire episode length which is the maximum number of steps or actions allowed within a single episode of the reinforcement learning process. By calculating the *td_error* as the difference between the reward and the cumulative episode reward, we are trying to capture the discrepancy between the immediate feedback received and the overall performance over an extended period. The latent input is then updated by adding a scaled noise term to introduce randomness and exploration. The scaling factor is determined by α, *td_error*, and the γ factor raised to the power of the current step:

$$\alpha \cdot td\_error \cdot \gamma^{step} \cdot latent\_input. \tag{5.1}$$

This update process helps the generator adjust its latent input based on the reward signal and explore different regions of the latent space. After the episode, the generator is updated using the final latent input. This scaling factor influences the magnitude of the noise added to the latent input, potentially increasing or decreasing the level of exploration based on the *td_error*'s magnitude. By scaling the noise with the *td_error*, the agent can adjust the exploration level dynamically during the training process. Higher *td_error* may correspond to larger scaling factors, leading to more exploration and increased randomness in the latent input updates. Conversely, lower *td_error* may result in smaller scaling factors, reducing the level of exploration and increasing the exploit as the agent refines its estimates and converges towards better solutions, this will decrease randomness in the latent input updates. Also by raising γ to the power of the current step, we apply a temporal discounting factor that decreases the impact of future rewards as the number of steps increases. As the agent gains more experience and learns from previous steps, the influence of future rewards on the scaling factor decreases, allowing the agent to focus more on optimizing its policy based on immediate feedback. The surrogate model evaluates the generator's output, and a target label tensor is created for the loss calculation. The generator's loss is computed using the cross-entropy loss function, and backward propagation is performed to update the generator's parameters. The trained generator is returned once the specified number of episodes is reached. In summary, this reinforcement learning approach in the GAN training process utilizes rewards from the surrogate model to guide the generator's learning. The latent input of the generator is updated using a noisy term scaled by the reward and reinforcement learning parameters. Iteratively optimizing the generator based on the reward signals aims to generate samples that can effectively deceive the surrogate model.

The physical implementation of the device follows the steps detailed in Chapter 4. Since training is done in advance, timings are not a matter of concern in this scenario, also considering the availability of cloud GPUs that can be rented cheaply.

5.4 Black Box Scenario

In the BB scenario, the attacker has no knowledge of the legitimate driving behavior nor the authenticator model architecture. Given the particularly challenging nature of this attack, the best course of action for a malicious party is to gain insight into one of the two factors to fall again into one of the two Gray Box scenarios. Thus, two approaches can be followed.

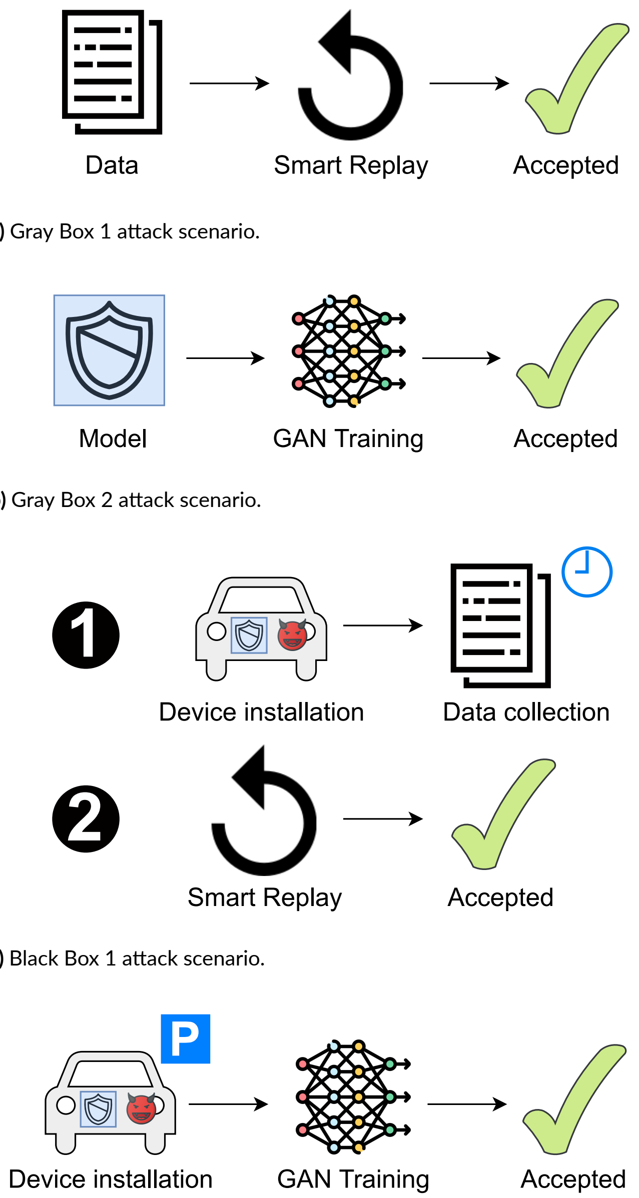
5.4.1 Gaining knowledge on the data (BB1)

To obtain legitimate driving behavior data, the attacker must use a two-stages attack, as summarized in Figure 5.1c. In the first stage ❶, the attacker identifies the target vehicle and physically deploys the malicious device connected to the CAN bus of the victim vehicle. It can be done, for instance, under the car's front bumper [6]. To ensure that the acquired data reaches the attacker, the malicious device needs to be equipped with telecommunication capabilities. This allows the device to transmit the sniffed CAN bus traffic to a remote location controlled by the attacker, enabling them to gather the collected data conveniently and discretely.

Once the attacker has amassed a sufficient amount of legitimate driving behavior data through this eavesdropping process, the second phase ❷ is the same as detailed for GB1.

5.4.2 Gaining knowledge on the model (BB2)

In this particular approach, which is depicted in Figure 5.1d, the attacker adopts a clever strategy to steal the car in a single stage, but at the expense of waiting longer to gain authentication to the vehicle. Indeed, to perform this attack, the thief exploits the implementation of the authenticator system while the vehicle is in an idle state. As explained in Chapter 3.2.3, when the car is parked or not moving, the authenticator model ignores any prediction for notification or countermeasures since the extracted features might lead to errors and false positives. Thus, the attacker can exploit this behavior to train a generator using the authenticator system as an oracle without risking alerting the vehicle owner. The generator is the same as the GB2 scenario. Indeed, by using the reinforcement learning procedure, the generator model can converge in just a few episodes. Once trained, the model can be used as a generator to forge packets to inject in the CAN bus and successfully steal the car.



(a) Gray Box 1 attack scenario.

(b) Gray Box 2 attack scenario.

(c) Black Box 1 attack scenario.

(d) Black Box 2 attack scenario.

Figure 5.1: Schema of the main steps for each attack.

6
Evaluation

We now evaluate our implemented behavior-based authentication system and our GAN-CAN attack on it. Our evaluation comprehends all scenarios detailed in the previous Chapters. We use three different metrics of evaluation: accuracy, F1-score, and Attack Success Rate (ASR). The ASR value, as perceived by the attacker, is equivalent to the False Acceptance Rate (FAR) value from the victim's standpoint. By using True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN), those metrics are formally defined as follows.

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}, \tag{6.1}$$

$$F1 = \frac{2TP}{2TP + FP + FN}, \tag{6.2}$$

$$ASR = \frac{\text{\# malicious batches fooling the authenticator}}{\text{\# malicious batches sent}}. \tag{6.3}$$

We start by evaluating the implementation of our authenticator in Chapter 6.1, which will serve as a baseline for our attacks. The evaluation of the GAN-CAN implementations is disclosed in Chapter 6.2, in which we treat each scenario separately.

Model	Accuracy	F1 Score
LSTM-FCN [78]	0.911	0.911
LSTM [23]	0.965	0.944
RNN-GRU [88]	0.957	0.960
Ensemble	**0.967**	**0.968**

Table 6.1: Baseline results of the authentication systems on the OCSLab dataset [4].

6.1 Baseline Authentication Results

After preprocessing the data and training our baseline models, as depicted in Chapter 3.2, we evaluated their performance on the test set. The observed results are shown in Table 6.1. To further improve the performance and exploit the strengths of these individual models, we created an ensemble by combining their predictions. Leveraging the diversity and complementary strengths of the individual models, the ensemble achieves enhanced prediction capabilities, with an accuracy of 0.967 on the test set. For our task, we divided the dataset into training, validation, and test sets with respective sizes of 85%, 5%, and 10% of the total dataset size. While keeping the window size at 16 and batch size at 4, we trained them for 120 epochs with a learning rate of 0.001.

6.2 GAN-CAN Results

To assess the effectiveness of the GAN-CAN framework, we can focus our evaluation on the GB1 and GB2 scenarios. Indeed, as stated in Chapter 4, in the WB scenario, the attacker has access to both data and model and can thus apply one of the gray boxes methodologies. Furthermore, the two black-box scenarios, BB1 and BB2, use the same approaches as GB1 and GB2, respectively. Here, the only difference resides in the data collection procedure for BB1 and the slower training procedure for BB2.

6.2.1 GB1 Evaluation

To evaluate the GB1 scenario, we consider each driver both as an attacker and a victim. For each attacker batch of data, we overwrite the *modifiable* features with the ones contained in the victim driver's batches (target). We then feed these combined batches of data to the ensemble model of the baseline authenticator. The purpose of this evaluation is to determine if the ensemble model can classify the combined data (*modifiable* features from the target driver data, while *borderline* and *non-modifiable* features from the attacker's driving behavior data) as the same class as the target driver. The experiments' results are shown in Figure 6.1. Each cell in the Figure has been evaluated on approximately 32 batches of data, which accounts for 2048 points in the dataset. The mean ASR obtained in all combinations is 0.865. This indicates that the *modifiable* features alone are sufficient for the attacker to deceive the ensemble of baseline authenticators and be classified as an authentic user. Therefore, an attacker can replay the modified data containing only modifiable features and still achieve successful authentication without compromising the driver's safety. However, it can be noticed that some of the drivers are more vulnerable than others (e.g., driver 0 and driver 4). This can be due to several factors, such as peculiar driving patterns in the training data or biases in the dataset creation (e.g., different routes, different times of collection). Conversely, some drivers (e.g., driver 6) appear more resilient to the attack, and thus relying on only *modifiable* features might not be enough to obtain a high ASR.

6.2.2 GB2 Evaluation

For the evaluation of GB2, we employ the generator model optimized through reinforcement learning, which utilizes the output of the ensemble of three models as the surrogate model for each driver in the dataset. Our goal is to generate data packages that could be classified as the authenticated driver. For the sake of this evaluation, we will consider each driver in the dataset as the legitimate driver once. We trained the generator to forge data specifically tailored to each driver in the dataset. Furthermore, as discussed comprehensively in Chapter 5.1, we use the generated data only for the *modifiable* features, while the others are extracted from the attacker's driving behavior. In our evaluation, we observe that the generator can successfully forge data for nine out of ten drivers in the dataset with a mean ASR of 0.994 and a standard deviation of 0.012. The training process of the generator involves randomness, and the output labels from the generator may have multiple possibilities. Consequently, there is a low chance that the generator may not converge, as it may not obtain the desired output label from the authentication model for generating data, but in our observations, we found that this issue of the generator not converging did not occur frequently in our experiments. Also, given that the training process takes a really short time , our attackers can easily initiate the training process again if needed, furthermore the timing is not a concern in GB2 scenario as the training is done offline. Upon careful analysis, we discovered that the generator's inability to converge for the problematic driver (driver 9) even after re-initiating the generator was related to the

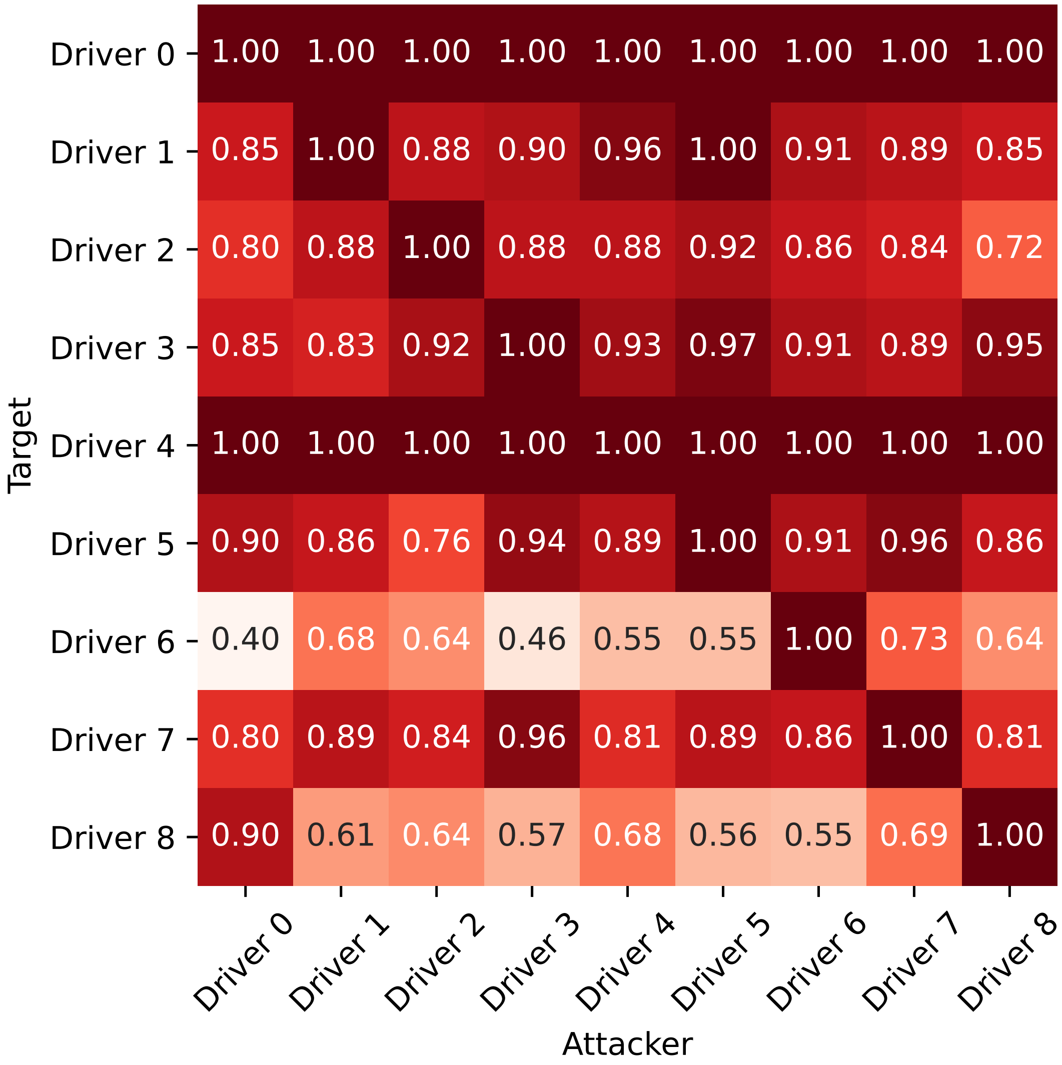


Figure 6.1: ASR for each target driver and each attacker driver for the GB1 scenario.

driver's feature class characteristics. It has the lowest representation within the dataset, and the authentication models struggled to train effectively in that class. As a result, the models do not produce accurate outputs for that class. In real-world usage, the authenticator would require the legitimate driver to provide more data (i.e., driving a little bit longer) for training. Since the generator relied on the output of the ensemble of models, it was unable to access reliable information for generating data specific to that driver. For this reason we did not consider driver 9 in our reported ASR. Although the forged data was created using the ensemble's output, it still was able to deceive each of the three baseline models and could be classified as the label of the targeted driver.

Nevertheless, these evaluations underscore the effectiveness of our generator model optimized through reinforcement learning in generating malicious data that can bypass the authentication models, even when limited to modifying only specific features.

6.2.3 BB1 Evaluation

In this scenario, we perform the same attack described in the GB1 scenario. However, the capability of this attack is limited by the data collection part, as shown in the first stage ❶ of Figure 5.1c. In the GB1 evaluation, we used the whole test set (for each driver individually) to calculate the ASR of the attack. Thus, since, on average, the dataset contains 9438 data points for each driver, and the test set size has been set to 10% of the total dataset size, the test set includes around 16 minutes of driving data for each driver. To evaluate the effects of smaller test sets, we consider different percentages of the original test set (from 10% to 90%, with a step size of 10%) and repeat the evaluation performed for GB1. The results show that the test set size does not affect the attack significantly, with a mean ASR of 0.894 and a deviation of 0.011. Therefore, the attacker can stop the first part of the attack after collecting just one batch of legitimate data and use that for all the following smart-replay attacks.

6.2.4 BB2 Evaluation

In this scenario, we use the same model as for the GB2 scenario. However, the training process is restricted by the authentication system behavior, which can generate only one prediction for each batch. Thus, the number of episodes the generator model needs to converge becomes tightly related to the time required to steal the car. Indeed, for each episode, we can generate only one batch of data, which is collected by the authentication system over 40 seconds. In Figure 6.2, we show the episode of convergence for the generator model while targeting each

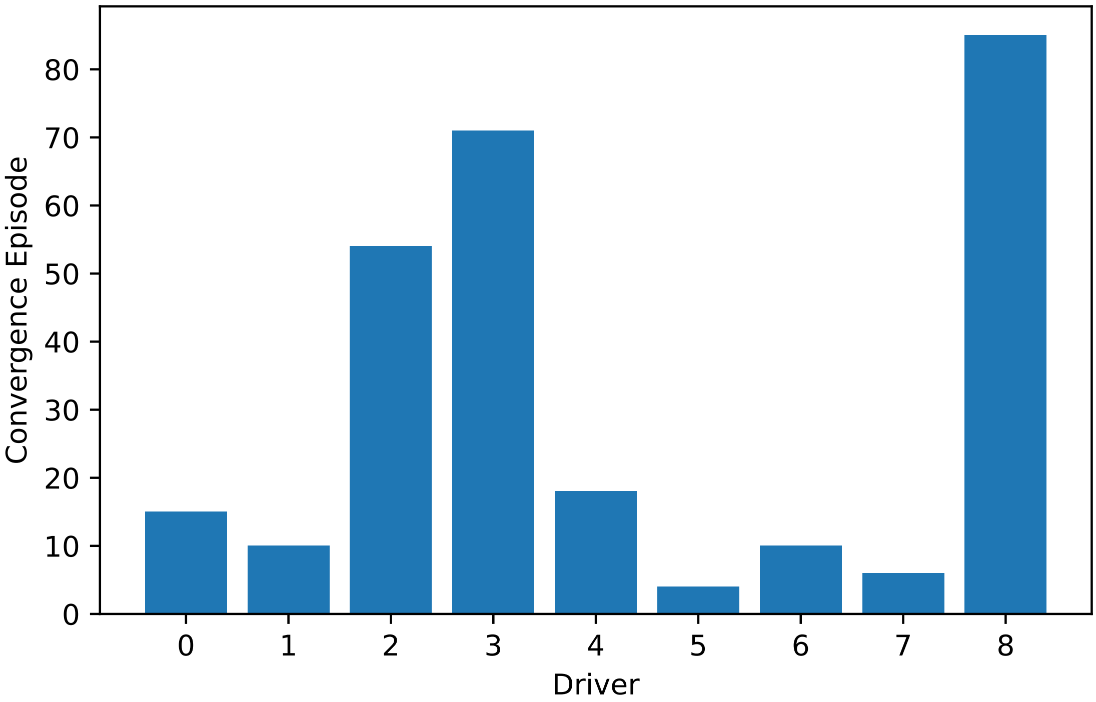


Figure 6.2: Episode of convergence of the generator model for each driver.

driver. The mean convergence episode is 30. The time needed for each episode to be processed, even on the CPU, is negligible (around 0.1 seconds). Therefore, the attacker will need, on average, 20 minutes to train the generator and then be able to inject malicious packets to fool the authenticator model. This restriction imposed by the authentication system behavior is pivotal in understanding the practical implications and time constraints associated with the attack. The inability to influence or accelerate the authentication system's data gathering process places limitations on the attack's speed. If the limitation on data gathering did not exist, the attacker would have had the opportunity to provide their generated data to the authentication system and obtain results much more quickly. Without the constraint of the fixed 40-second data collection per batch, the attacker could potentially accelerate the process. With this increased efficiency, the attacker would have been able to generate batches of legitimate data in under two minute, significantly reducing the time required to execute the attack. However, it is crucial to recognize that such a scenario assumes an unrestricted and idealized setup, which is not reflective of real-world constraints and practical implementations that we suggested in Chapter 3.2.2, so we will not consider this scenario in our evaluation.

	ASR	Time to steal the car			
		Setup	Data	Training	Total
WB	0.994	2 min	-	*offline*	**2 min**
GB1	0.865	2 min	-	-	**2 min**
GB2	0.994	2 min	-	*offline*	**2 min**
BB1	0.865	2 min	1 min	-	**3 min**
BB2	0.994	2 min	-	20 min	**22 min**

Table 6.2: Summary of the GAN-CAN results.

6.3 Summary

Throughout the thesis, the central objective of an attacker is to gain unauthorized access and ultimately steal a vehicle. With this critical goal in mind, the thesis extensively explores various strategies and tactics that adversaries may employ based on their level of knowledge. In the WB scenario, all the attacks are doable. However, a smart attacker will use the easiest and faster attack (i.e., GB2). As summarized in Table 6.2, for an attacker without any knowledge of the victim's system, the best solution depends on the attacker's access to the victim's vehicle. If they have access to it more than once, BB1 is the stealthy way to steal the car. They can also insert a GPS tracker in the malicious device to easily find the vehicle for the second stage. In that scenario, the time for data collection must also be taken into account. However, since we show that the attack is successful even with only one batch of legitimate data, this process should take at least 1 minute. More data could be needed to emulate the driver behavior, for instance, to try to fool an IDS. Instead, for GB1 and GB2, this procedure is not necessary since, in the former, data knowledge is part of the assumptions, and in the latter, data is generated from randomness through the generator model. Otherwise, an attacker can leverage BB2 during nighttime or while the victim is away from the vehicle for some time (22 minutes). However, independently of the scenario, an attacker can successfully steal the car in a reasonable time. The feasibility of stealing the car within a reasonable time emphasizes the importance of continuous improvement and vigilance in developing robust defense mechanisms. Researchers and developers must stay one step ahead of potential adversaries, continually updating the authentication system's security measures to counter evolving attack strategies.

7 Countermeasures

In this thesis, we discussed different attacks on state-of-the-art behavior-based authentication systems, showing that even the best solution in the literature can be beaten by GAN-CAN. To mitigate our attacks, some countermeasures can be employed. One of the root causes of the vulnerability is the unauthenticated and unencrypted nature of the CAN bus. Some research proposes to secure the communication with lightweight encryption [111] or authentication measures [112]. However, an attacker can still replay packets if no proper sequence number verification is done or can understand the content of the messages by exploiting encrypted packet analysis [113]. Moreover, these solutions can add overhead on a safety-critical channel, exposing the driver to safety risks. Another approach can be the usage of IDSs to identify unusual patterns in the CAN bus traffic. By analyzing the driver's behavior and detecting deviations from normal patterns, potential replay attacks can be detected and flagged for further investigation [114]. For example, deploying an IDS system in the CAN bus can affect the data collection phase of BB1. Indeed, while we show how an attacker needs only one batch of data to perform the attack, replaying the same data samples repeatedly might alert the system. Thus, the attacker might need to collect more data to fool both the authenticator and the IDS system.

Other approaches can be considered in the behavior-based authentication system as well. Protecting the model used for authentication is crucial to prevent attackers from accessing and analyzing its output. Techniques such as model encryption, obfuscation, and secure model deployment can be employed to safeguard the model's integrity and confidentiality [115, 116].

Moreover, incorporating adversarial training techniques during the model training process can enhance its robustness against attacks [117]. By exposing the model to adversarial examples during training, it can learn to better differentiate between genuine and malicious inputs, making it more difficult for attackers to forge data. In our pursuit of countering the GAN-CAN attack, we explored the effectiveness of a simple adversarial training approach. Our aim was to generate a sufficient amount of malicious data for all classes in the dataset. Subsequently, we labeled these newly generated malicious data and merged them with the authenticated dataset. Our intention was to train the authentication models using this augmented dataset, with the hope that they would be able to successfully differentiate between the generated malicious data and the authentic data. However, the performance of the authentication models significantly dropped after training with the augmented dataset. This outcome indicates that the some part of generated malicious data bears a resemblance to the authentic data. The similarity between the two types of data posed a challenge for the authentication models, making it difficult for them to accurately distinguish between genuine and malicious instances.

This result sheds light on the resilience of the GAN-CAN attack and highlights the sophistication of the generated malicious data. The similarities between the generated and authentic data, despite efforts to label and combine them, indicate that a more comprehensive and sophisticated defense mechanism is necessary to effectively counter the GAN-CAN attack.

Our findings emphasize the need for further research and exploration of advanced countermeasures beyond simple adversarial training. These countermeasures should take into account the unique characteristics and challenges posed by GAN-CAN, aiming to develop robust authentication models that can withstand the increasingly sophisticated attacks in the realm of behavior-based driver authentication systems.

7.1 Does Behavior-Based Authentication alone provide an effective solution ?

Behavior-based authentication is undoubtedly a valuable component in the authentication process, contributing to a higher level of security. However, relying solely on this method might leave potential vulnerabilities in the system. To bolster the security further, adopting a multi-layered approach that incorporates additional authentication methods is recommended. This makes it much more difficult for attackers to gain access, even if they are able to defeat one of the authentication methods. Integrating biometrics, passwords, or other authentication alongside behavior-based authentication can create a robust and dynamic security framework. Each layer contributes unique strengths to the overall authentication process, making it more difficult for attackers to compromise the system. Nevertheless, a balance must be struck between security and usability. Introducing multiple layers of security may enhance protection, but it could also lead to user inconvenience and annoyance. The challenge lies in finding the right balance that maintains a high level of security without compromising user experience. Thus, while eliminating the possibility of the GAN-CAN attack is desirable, the design of the authentication system should be carefully considered to achieve an optimal trade-off between security and usability. It requires a thoughtful approach to design an authentication system that seamlessly integrates security layers while minimizing disruptions to the driver experience. Implementing strong security measures without unduly burdening the driver is also crucial for widespread adoption and seamless user experience. Moreover implementing transparent and efficient authentication mechanisms can help drivers feel more at ease while ensuring their safety.

8

Conclusion and future works

8.1 Conclusion

Despite the huge interest in finding novel solutions to prevent car theft and to provide novel authentication factors, we demonstrated the fragility of the behavior-based authentication system for drivers. While these systems have been researched for many years, their practical deployment is still problematic. Indeed, we identified possible implementations of these systems and highlighted their potential vulnerabilities. After developing the best behavior-based authentication solution starting from the state-of-the-art, we demonstrated how a properly trained GAN could defeat such a system under various hypotheses on the adversary's knowledge and capabilities with a high success rate in a few minutes.

However, with the advancements of DL, better systems may be created based on driver behaviors. The findings of this study underscore the need for robust security measures to protect against attacks on behavior-based authentication systems in vehicles. Future research should focus on developing enhanced defense mechanisms and secure implementations to mitigate the risks associated with these emerging authentication approaches. The automotive industry must also address the practical challenges of implementing such complex security systems in vehicles without compromising user convenience or overburdening drivers with cumbersome processes. Striking the right balance between robust security and user experience will be a critical aspect of shaping the future of automotive authentication.

Furthermore, GAN-CAN, has provided a promising tool for researchers to evaluate the security of newly proposed behavior-based authentication models against adversary attacks.

8.2 Future works

8.2.1 Refining and improving GAN-CAN

Further research can focus on refining the GAN-CAN attack to enhance its effectiveness and success rate while minimizing the time required for deployment. This can involve exploring different variations of generator architectures, optimization techniques, and injection strategies. Additionally, the research can extend to studying the transferability of the GAN-CAN attack across different behavior-based authentication models. Assessing the attack's performance on various systems can reveal common vulnerabilities and provide insights into improving the overall security of behavior-based authentication across the automotive industry.

8.2.2 Exploring countermeasures

As highlighted before, there is an urgent need to address the security vulnerabilities in behavior-based driver authentication systems. Future work can involve investigating and proposing effective countermeasures to mitigate the GAN-CAN attack. This may include enhancing the security of the CAN, implementing IDSs, or developing more robust authentication mechanisms that combine behavior-based analysis with additional layers of verification.

8.2.3 Assessing the impact on real-world scenarios

It would be valuable to assess the impact of the GAN-CAN attack in real-world scenarios by considering factors such as different driving conditions, variations in sensor data quality, and diverse vehicle models. Considering diverse vehicle models in the evaluation of the GAN-CAN attack offers valuable insights into its adaptability across various automotive platforms. It helps identify potential weaknesses in specific vehicle models and guides manufacturers in fortifying their systems against the attack.

8.2.4 Ethical considerations and responsible disclosures

Future work should also address ethical considerations related to the GAN-CAN attack. This involves analyzing the potential consequences of such attacks, understanding the legal and privacy implications, and developing responsible disclosure guidelines to ensure the findings are appropriately communicated to relevant stakeholders, including vehicle manufacturers and security researchers. Analyzing the potential consequences involves evaluating the impact of successful GAN-CAN attacks on vehicle security, safety, and privacy. This assessment should extend to the broader implications, such as the potential for accidents, theft, or even life-threatening situations.

8.2.5 Testing with bigger datasets

To further evaluate the resilience and adaptability of behavior-based authentication systems against the GAN-CAN attack, future work should involve testing with larger datasets. This will provide a more comprehensive assessment of the attack's success rate, its ability to generate realistic malicious data, and the impact on the authentication models' performance.

By focusing on these areas of future work, we can further advance the understanding of behavior-based driver authentication vulnerabilities and contribute to the development of robust and secure systems for vehicle security.

References

[1] R. Islam and R. U. D. Refat, "Improving can bus security by assigning dynamic arbitration ids," 2020, pp. 1–13.

[2] M. Bozdal, M. Samie, S. Aslam, and I. Jennions, "Evaluation of can bus security challenges," *Sensors*, vol. 20, no. 8, p. 2364, 2020.

[3] K. Tindell, "A complete can bus frame," 2020, cC BY-SA 4.0 License. [Online]. Available: https://kentindell.github.io/assets/images/fixed-wiki-can-frame.png

[4] F. Martinelli, F. Mercaldo, A. Orlando, V. Nardone, A. Santone, and A. K. Sangaiah, "Human behavior characterization for driving style recognition in vehicle system," *Computers & Electrical Engineering*, vol. 83, p. 102504, 2020.

[5] F. D. Garcia, D. F. Oswald, T. Kasper, and P. Pavlidès, "Lock it and still lose it-on the (in) security of automotive remote keyless entry systems." in *USENIX security symposium*, vol. 53, 2016.

[6] K. Tindell, "Can injection: keyless car theft," https://kentindell.github.io/2023/04/03/can-injection/, April 2023.

[7] E. Derman and A. A. Salah, "Continuous real-time vehicle driver authentication using convolutional neural network based face recognition," in *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, 2018, pp. 577–584.

[8] H. Moon and K. Lee, "Biometric driver authentication based on 3d face recognition for telematics applications," in *Universal Acess in Human Computer Interaction. Coping with Diversity*, C. Stephanidis, Ed. Springer Berlin Heidelberg, 2007, pp. 473–480.

[9] S. Gupta, A. Buriro, and B. Crispo, "Driverauth: A risk-based multi-modal biometric-based driver authentication scheme for ride-sharing platforms," vol. 83, 2019, pp. 122–139. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404818310113

[10] I. Nakanishi, S. Baba, and S. Li, “Evaluation of brain waves as biometrics for driver authentication using simplified driving simulator,” in *2011 International Conference on Biometrics and Kansei Engineering*, 2011, pp. 71–76.

[11] T. Scheidat, M. Biermann, J. Dittmann, C. Vielhauer, and K. Kümmel, “Multi-biometric fusion for driver authentication on the example of speech and face,” in *Biometric ID Management and Multimodal Communication*, J. Fierrez, J. Ortega-Garcia, A. Esposito, A. Drygajlo, and M. Faundez-Zanuy, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 220–227.

[12] B. Taha, S. N. A. Seha, D. Y. Hwang, and D. Hatzinakos, “Eyedrive: A deep learning model for continuous driver authentication,” vol. 17, no. 3, 2023, pp. 637–647.

[13] Y. Xun, J. Liu, N. Kato, Y. Fang, and Y. Zhang, “Automobile driver fingerprinting: A new machine learning based authentication scheme,” vol. 16, no. 2, 2020, pp. 1417–1426.

[14] N. Abdennour, T. Ouni, and N. B. Amor, “Driver identification using only the can-bus vehicle data through an rcn deep learning approach,” *Robotics and Autonomous Systems*, vol. 136, p. 103707, 2021.

[15] C. Ravi, A. Tigga, G. T. Reddy, S. Hakak, and M. Alazab, “Driver identification using optimized deep learning model in smart transportation,” *ACM Transactions on Internet Technology*, vol. 22, no. 4, pp. 1–17, 2022.

[16] C. Miller and C. Valasek, “Remote exploitation of an unaltered passenger vehicle,” *Black Hat USA*, vol. 2015, no. S 91, pp. 1–91, 2015.

[17] A. Cura, H. Küçük, E. Ergen, and I. B. Öksüzoğlu, “Driver Profiling Using Long Short Term Memory (LSTM) and Convolutional Neural Network (CNN) Methods,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 10, pp. 6572–6582, Oct. 2021, conference Name: IEEE Transactions on Intelligent Transportation Systems.

[18] G. A. M. Meiring and H. C. Myburgh, “A Review of Intelligent Driving Style Analysis Systems and Related Artificial Intelligence Algorithms,” *Sensors*, vol. 15, no. 12, pp. 30 653–30 682, Dec. 2015, number: 12 Publisher: Multidisciplinary Digital Publishing Institute. [Online]. Available: https://www.mdpi.com/1424-8220/15/12/29822

[19] D. Jeong, M. Kim, K. Kim, T. Kim, J. Jin, C. Lee, and S. Lim, "Real-time driver identification using vehicular big data and deep learning," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 123–130.

[20] M. N. Azadani and A. Boukerche, "Driver identification using vehicular sensing data: A deep learning approach," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2021, pp. 1–6.

[21] A. Burton, T. Parikh, S. Mascarenhas, J. Zhang, J. Voris, N. S. Artan, and W. Li, "Driver identification and authentication with active behavior modeling," in *2016 12th International Conference on Network and Service Management (CNSM)*. IEEE, 2016, pp. 388–393.

[22] X. Lin, K. Zhang, W. Cao, and L. Zhang, "Driver evaluation and identification based on driving behavior data," in *2018 5th International Conference on Information Science and Control Engineering (ICISCE)*. IEEE, 2018, pp. 718–722.

[23] A. Girma, X. Yan, and A. Homaifar, "Driver identification based on vehicle telematics data using lstm-recurrent neural network," in *2019 IEEE 31st International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 2019, pp. 894–902.

[24] B. Biggio, I. Corona, D. Maiorca, B. Nelson, N. Šrndić, P. Laskov, G. Giacinto, and F. Roli, "Evasion attacks against machine learning at test time," in *Machine Learning and Knowledge Discovery in Databases: European Conference, ECML PKDD 2013, Prague, Czech Republic, September 23-27, 2013, Proceedings, Part III 13*. Springer, 2013, pp. 387–402.

[25] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," in *Proceedings of the 29th International Coference on International Conference on Machine Learning*, ser. ICML'12. Madison, WI, USA: Omnipress, 2012, p. 1467–1474.

[26] A. Demontis, M. Melis, M. Pintor, M. Jagielski, B. Biggio, A. Oprea, C. Nita-Rotaru, and F. Roli, "Why do adversarial attacks transfer? explaining transferability of evasion and poisoning attacks," in *28th USENIX security symposium (USENIX security 19)*, 2019, pp. 321–338.

[27] W. Hu and Y. Tan, "Generating adversarial malware examples for black-box attacks based on gan," in *Data Mining and Big Data: 7th International Conference, DMBD 2022, Beijing, China, November 21–24, 2022, Proceedings, Part II*. Springer, 2023, pp. 409–423.

[28] C. Corbett, E. Schoch, F. Kargl, and F. Preussner, "Automotive ethernet: Security opportunity or challenge?" *Sicherheit 2016-Sicherheit, Schutz und Zuverlässigkeit*, 2016.

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

[30] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Communications of the ACM*, vol. 63, no. 11, pp. 139–144, 2020.

[31] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, "Generative adversarial networks: An overview," vol. 35, no. 1, 2018, pp. 53–65.

[32] Y.-J. Cao, L.-L. Jia, Y.-X. Chen, N. Lin, C. Yang, B. Zhang, Z. Liu, X.-X. Li, and H.-H. Dai, "Recent advances of generative adversarial networks in computer vision," *IEEE Access*, vol. 7, pp. 14 985–15 006, 2018.

[33] D. Croce, G. Castellucci, and R. Basili, "Gan-bert: Generative adversarial learning for robust text classification with a bunch of labeled examples," in *Proceedings of the 58th annual meeting of the association for computational linguistics*, 2020, pp. 2114–2119.

[34] J. Zhang, J. Chen, D. Wu, B. Chen, and S. Yu, "Poisoning attack in federated learning using generative adversarial nets," in *2019 18th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/13th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE)*, 2019, pp. 374–380.

[35] Y. Liang, S. Samtani, B. Guo, and Z. Yu, "Behavioral biometrics for continuous authentication in the internet-of-things era: An artificial intelligence perspective," *IEEE Internet of Things Journal*, vol. 7, no. 9, pp. 9128–9143, 2020.

[36] H. Jagadeesan and M. S. Hsiao, "A novel approach to design of user re-authentication systems," in *2009 IEEE 3rd International Conference on Biometrics: Theory, Applications, and Systems*. IEEE, 2009, pp. 1–6.

[37] S. P. Banerjee and D. L. Woodard, "Biometric authentication and identification using keystroke dynamics: A survey," *Journal of Pattern Recognition Research*, vol. 7, no. 1, pp. 116–139, 2012.

[38] C. Shen, Z. Cai, X. Guan, Y. Du, and R. A. Maxion, "User authentication through mouse dynamics," *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 1, pp. 16–30, 2012.

[39] B. Draffin, J. Zhu, and J. Zhang, "Keysens: Passive user authentication through micro-behavior modeling of soft keyboard interaction," in *Mobile Computing, Applications, and Services: 5th International Conference, MobiCASE 2013, Paris, France, November 7-8, 2013, Revised Selected Papers 5*. Springer, 2014, pp. 184–201.

[40] M. Shahzad, A. X. Liu, and A. Samuel, "Behavior based human authentication on touch screen devices using gestures and signatures," *IEEE Transactions on Mobile Computing*, vol. 16, no. 10, pp. 2726–2741, 2016.

[41] A. B. A. Ali, V. Ponnusamy, A. Sangodiah, R. Alroobaea, N. Jhanjhi, U. Ghosh, and M. Masud, "Smartphone security using swipe behavior-based authentication," *Intelligent Automation & Soft Computing*, vol. 29, no. 2, pp. 571–585, 2021.

[42] R. Miller, N. K. Banerjee, and S. Banerjee, "Within-system and cross-system behavior-based biometric authentication in virtual reality," in *2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW)*. IEEE, 2020, pp. 311–316.

[43] R. Miller, A. Ajit, N. K. Banerjee, and S. Banerjee, "Realtime behavior-based continual authentication of users in virtual reality environments," in *2019 IEEE International Conference on Artificial Intelligence and Virtual Reality (AIVR)*. IEEE, 2019, pp. 253–2531.

[44] M. Villa, M. Gofman, and S. Mitra, "Survey of biometric techniques for automotive applications," in *Information Technology-New Generations: 15th International Conference on Information Technology*. Springer, 2018, pp. 475–481.

[45] Q. Wang and S. Sawhney, "Vecure: A practical security framework to protect the can bus of vehicles," in *2014 International Conference on the Internet of Things (IOT)*, 2014, pp. 13–18.

[46] K. Iehira, H. Inoue, and K. Ishida, "Spoofing attack using bus-off attacks against a specific ecu of the can bus," in *2018 15th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. IEEE, 2018, pp. 1–4.

[47] G. Bloom, "Weepingcan: A stealthy can bus-off attack," in *Workshop on Automotive and Autonomous Vehicle Security*, 2021.

[48] L. Dariz, M. Selvatici, M. Ruggeri, G. Costantino, and F. Martinelli, "Trade-off analysis of safety and security in can bus communication," in *2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*, 2017, pp. 226–231.

[49] S. Fassak, Y. El Hajjaji El Idrissi, N. Zahid, and M. Jedra, "A secure protocol for session keys establishment between ecus in the can bus," in *2017 International Conference on Wireless Networks and Mobile Communications (WINCOM)*, 2017, pp. 1–6.

[50] W. A. Farag, "Cantrack: Enhancing automotive can bus security using intuitive encryption algorithms," in *2017 7th International Conference on Modeling, Simulation, and Applied Optimization (ICMSAO)*, 2017, pp. 1–5.

[51] K. Cheng, Y. Bai, Y. Zhou, Y. Tang, D. Sanan, and Y. Liu, "Caneleon: Protecting can bus with frame id chameleon," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 7, pp. 7116–7130, 2020.

[52] J. Halabi and H. Artail, "A lightweight synchronous cryptographic hash chain solution to securing the vehicle can bus," in *2018 IEEE International Multidisciplinary Conference on Engineering Technology (IMCET)*, 2018, pp. 1–6.

[53] S. Checkoway, D. McCoy, B. Kantor, D. Anderson, H. Shacham, S. Savage, K. Koscher, A. Czeskis, F. Roesner, T. Kohno *et al.*, "Comprehensive experimental analyses of automotive attack surfaces." in *USENIX security symposium*, vol. 4, no. 447-462. San Francisco, 2011, p. 2021.

[54] I. Rouf, R. D. Miller, H. A. Mustafa, T. Taylor, S. Oh, W. Xu, M. Gruteser, W. Trappe, and I. Seskar, "Security and privacy vulnerabilities of in-car wireless networks: A tire pressure monitoring system case study." in *USENIX Security Symposium*, vol. 10, 2010.

[55] S.-F. Lokman, A. T. Othman, and M.-H. Abu-Bakar, "Intrusion detection system for automotive controller area network (can) bus system: a review," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, pp. 1–17, 2019.

[56] 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)*, 2016, pp. 176–180.

[57] M. Casillo, S. Coppola, M. De Santo, F. Pascale, and E. Santonicola, "Embedded intrusion detection system for detecting attacks over can-bus," in *2019 4th International Conference on System Reliability and Safety (ICSRS)*, 2019, pp. 136–141.

[58] B. Lampe and W. Meng, "Ids for can: A practical intrusion detection system for can bus security," in *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*, 2022, pp. 1782–1787.

[59] S. Jin, J.-G. Chung, and Y. Xu, "Signature-based intrusion detection system (ids) for in-vehicle can bus network," in *2021 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2021, pp. 1–5.

[60] R. Islam, M. K. Devnath, M. D. Samad, and S. M. Jaffrey Al Kadry, "Ggnb: Graph-based gaussian naive bayes intrusion detection system for can bus," vol. 33, 2022, p. 100442. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S221420962100111X

[61] D. Caivano, M. De Vincentiis, F. Nitti, and A. Pal, "Quantum optimization for fast can bus intrusion detection." New York, NY, USA: Association for Computing Machinery, 2022, p. 15–18. [Online]. Available: https://doi.org/10.1145/3549036.3562058

[62] 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)*, 2015, pp. 45–49.

[63] Q. Zhao, M. Chen, Z. Gu, S. Luan, H. Zeng, and S. Chakrabory, "Can bus intrusion detection based on auxiliary classifier gan and out-of-distribution detection," *ACM Trans. Embed. Comput. Syst.*, vol. 21, no. 4, 2022.

[64] S. Longari, D. H. N. Valcarcel, M. Zago, M. Carminati, and S. Zanero, "Cannolo: An anomaly detection system based on lstm autoencoders for controller area network," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1913–1924, 2020.

[65] M. D. Hossain, H. Inoue, H. Ochiai, D. Fall, and Y. Kadobayashi, "Lstm-based intrusion detection system for in-vehicle can bus communications," *IEEE Access*, vol. 8, pp. 185 489–185 502, 2020.

[66] I. Corona, G. Giacinto, and F. Roli, "Adversarial attacks against intrusion detection systems: Taxonomy, solutions and open issues," *Information Sciences*, vol. 239, pp. 201–225, 2013.

[67] M. Pawlicki, M. Choraś, and R. Kozik, "Defending network intrusion detection systems against adversarial evasion attacks," *Future Generation Computer Systems*, vol. 110, pp. 148–154, 2020.

[68] R. Chauhan and S. S. Heydari, "Polymorphic adversarial ddos attack on ids using gan," in *2020 International Symposium on Networks, Computers and Communications (ISNCC)*. IEEE, 2020, pp. 1–6.

[69] D. Shu, N. O. Leslie, C. A. Kamhoua, and C. S. Tucker, "Generative adversarial attacks against intrusion detection systems using active learning," in *Proceedings of the 2nd ACM Workshop on Wireless Security and Machine Learning*, ser. WiseML '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 1–6. [Online]. Available: https://doi.org/10.1145/3395352.3402618

[70] G. Apruzzese, M. Andreolini, L. Ferretti, M. Marchetti, and M. Colajanni, "Modeling realistic adversarial attacks against network intrusion detection systems," vol. 3, no. 3. New York, NY, USA: Association for Computing Machinery, feb 2022. [Online]. Available: https://doi.org/10.1145/3469659

[71] Z. Lin, Y. Shi, and Z. Xue, “Idsgan: Generative adversarial networks for attack generation against intrusion detection,” in *Advances in Knowledge Discovery and Data Mining*. Cham: Springer International Publishing, 2022, pp. 79–91.

[72] X. Zhou, W. Liang, W. Li, K. Yan, S. Shimizu, and K. I.-K. Wang, “Hierarchical adversarial attacks against graph-neural-network-based iot network intrusion detection system,” vol. 9, no. 12, 2022, pp. 9310–9319.

[73] H. Jiang, J. Lin, and H. Kang, “Fgmd: A robust detector against adversarial attacks in the iot network,” *Future Generation Computer Systems*, vol. 132, pp. 194–210, 2022.

[74] H. Mohammadian, A. A. Ghorbani, and A. H. Lashkari, “A gradient-based approach for adversarial attack on deep learning-based network intrusion detection systems,” *Applied Soft Computing*, vol. 137, p. 110173, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494623001916

[75] I. Debicha, B. Cochez, T. Kenaza, T. Debatty, J.-M. Dricot, and W. Mees, “Adv-bot: Realistic adversarial botnet attacks against network intrusion detection systems,” *Computers Security*, vol. 129, p. 103176, 2023.

[76] D. He, J. Dai, X. Liu, S. Zhu, S. Chan, and M. Guizani, “Adversarial attacks for intrusion detection based on bus traffic,” *IEEE Network*, vol. 36, no. 4, pp. 203–209, 2022.

[77] M. Usama, M. Asim, S. Latif, J. Qadir, and Ala-Al-Fuqaha, “Generative adversarial networks for launching and thwarting adversarial attacks on network intrusion detection systems,” in *2019 15th International Wireless Communications Mobile Computing Conference (IWCMC)*, 2019, pp. 78–83.

[78] A. El Mekki, A. Bouhoute, and I. Berrada, “Improving driver identification for the next-generation of in-vehicle software systems,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7406–7415, 2019.

[79] E. Erzin, Y. Yemez, A. M. Tekalp, A. Erçil, H. Erdogan, and H. Abut, “Multimodal person recognition for human-vehicle interaction,” *IEEE MultiMedia*, vol. 13, no. 2, pp. 18–31, 2006.

[80] U. Fugiglando, P. Santi, S. Milardo, K. Abida, and C. Ratti, “Characterizing the” driver dna” through can bus data analysis,” in *Proceedings of the 2nd ACM International Work-*

shop on Smart, Autonomous, and Connected Vehicular Systems and Services, 2017, pp. 37–41.

[81] U. Fugiglando, E. Massaro, P. Santi, S. Milardo, K. Abida, R. Stahlmann, F. Netter, and C. Ratti, "Driving behavior analysis through can bus data in an uncontrolled environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 737–748, 2018.

[82] B. I. Kwak, J. Woo, and H. K. Kim, "Know your master: Driver profiling-based anti-theft method," in *2016 14th Annual Conference on Privacy, Security and Trust (PST)*. IEEE, 2016, pp. 211–218.

[83] G. Ahmadi-Assalemi, H. M. Al-Khateeb, C. Maple, G. Epiphaniou, M. Hammoudeh, H. Jahankhani, and P. Pillai, "Optimising driver profiling through behaviour modelling of in-car sensor and global positioning system data," *Computers & Electrical Engineering*, vol. 91, p. 107047, 2021.

[84] M. A. Rahim, L. Zhu, X. Li, J. Liu, Z. Zhang, Z. Qin, S. Khan, and K. Gai, "Zero-to-stable driver identification: A non-intrusive and scalable driver identification scheme," *IEEE transactions on vehicular technology*, vol. 69, no. 1, pp. 163–171, 2019.

[85] S. Schneegass, B. Pfleging, N. Broy, F. Heinrich, and A. Schmidt, "A data set of real world driving to assess driver workload," in *Proceedings of the 5th international conference on automotive user interfaces and interactive vehicular applications*, 2013, pp. 150–157.

[86] M. N. Azadani and A. Boukerche, "Performance evaluation of driving behavior identification models through can-bus data," in *2020 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2020, pp. 1–6.

[87] H. Abu-Gellban, L. Nguyen, M. Moghadasi, Z. Pan, and F. Jin, "Livedi: An anti-theft model based on driving behavior," in *Proceedings of the 2020 ACM Workshop on Information Hiding and Multimedia Security*, 2020, pp. 67–72.

[88] B. Gahr, S. Liu, K. Koch, F. Barata, A. Dahlinger, B. Ryder, E. Fleisch, and F. Wortmann, "Driver identification via the steering wheel," *arXiv preprint arXiv:1909.03953*, 2019.

[89] J. Zhang, Z. Wu, F. Li, C. Xie, T. Ren, J. Chen, and L. Liu, "A deep learning framework for driving behavior identification on in-vehicle can-bus sensor data," *Sensors*, vol. 19, no. 6, p. 1356, 2019.

[90] Y. Xun, J. Liu, N. Kato, Y. Fang, and Y. Zhang, "Automobile driver fingerprinting: A new machine learning based authentication scheme," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 1417–1426, 2019.

[91] S. Jafarnejad, G. Castignani, and T. Engel, "Towards a real-time driver identification mechanism based on driving sensing data," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2017, pp. 1–7.

[92] J. Chen, Z. Wu, and J. Zhang, "Driver identification based on hidden feature extraction by using adaptive nonnegativity-constrained autoencoder," *Applied Soft Computing*, vol. 74, pp. 1–9, 2019.

[93] I. Del Campo, R. Finker, M. V. Martinez, J. Echanobe, and F. Doctor, "A real-time driver identification system based on artificial neural networks and cepstral analysis," in *2014 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2014, pp. 1848–1855.

[94] K. H. Park and H. K. Kim, "This car is mine!: Automobile theft countermeasure leveraging driver identification with generative adversarial networks," *arXiv preprint arXiv:1911.09870*, 2019.

[95] P.-Y. Tseng, P.-C. Lin, and E. Kristianto, "Vehicle theft detection by generative adversarial networks on driving behavior," *Engineering Applications of Artificial Intelligence*, vol. 117, p. 105571, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0952197622005619

[96] C. Miyajima, Y. Nishiwaki, K. Ozawa, T. Wakita, K. Itou, K. Takeda, and F. Itakura, "Driver modeling based on driving behavior and its evaluation in driver identification," *Proceedings of the IEEE*, vol. 95, no. 2, pp. 427–437, 2007.

[97] Y. G. Kang, K. H. Park, and H. K. Kim, "Automobile theft detection by clustering owner driver data," *arXiv preprint arXiv:1909.08929*, 2019.

[98] D. Hallac, A. Sharang, R. Stahlmann, A. Lamprecht, M. Huber, M. Roehder, J. Leskovec *et al.*, "Driver identification using automobile sensor data from a single turn," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2016, pp. 953–958.

[99] S. Ezzini, I. Berrada, and M. Ghogho, "Who is behind the wheel? driver identification and fingerprinting," *Journal of Big Data*, vol. 5, no. 1, pp. 1–15, 2018.

[100] L. Marchegiani and I. Posner, "Long-term driving behaviour modelling for driver identification," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 913–919.

[101] Linux Foundation Project, "Automotive grade linux: a fully open software stack for the connected car, with linux at its core." https://www.automotivelinux.org/, 2023.

[102] B. Gahr, B. Ryder, A. Dahlinger, and F. Wortmann, "Driver identification via brake pedal signals—a replication and advancement of existing techniques," in *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2018, pp. 1415–1420.

[103] Raspberry Pi Foundation, "Raspberry Pi 4: Your tiny, dual-display, desktop computer and robot brains, smart home hub, media centre, networked AI core, factory controller, and much more," https://www.raspberrypi.com/products/raspberry-pi-4-model-b/, 2023.

[104] J. Picard and B. J. Male, "System and method for detecting disconnection of a device," US Patent US20110154086A1, Jun., 2011. [Online]. Available: https://patents.google.com/patent/US20110154086/ru

[105] H. Sasahara, T. Ishizaki, J.-I. Imura, and H. Sandberg, "Disconnection-aware attack detection and isolation with separation-based detector reconfiguration," *IEEE Transactions on Control Systems Technology*, vol. 30, no. 4, pp. 1625–1640, 2022.

[106] commaai, "opendbc: democratize access to car decoder rings," https://github.com/commaai/opendbc, 2023.

[107] M. Jagielski, N. Carlini, D. Berthelot, A. Kurakin, and N. Papernot, "High accuracy and high fidelity extraction of neural networks," in *29th USENIX security symposium (USENIX Security 20)*, 2020, pp. 1345–1362.

[108] Y. Zhu, Y. Cheng, H. Zhou, and Y. Lu, "Hermes attack: Steal dnn models with lossless inference accuracy." in *USENIX Security Symposium*, 2021, pp. 1973–1988.

[109] X. Gong, Q. Wang, Y. Chen, W. Yang, and X. Jiang, "Model extraction attacks and defenses on cloud-based machine learning models," *IEEE Communications Magazine*, vol. 58, no. 12, pp. 83–89, 2020.

[110] X. Zhang, C. Fang, and J. Shi, "Thief, beware of what get you there: Towards understanding model extraction attack," *arXiv preprint arXiv:2104.05921*, 2021.

[111] Z. Lu, Q. Wang, X. Chen, G. Qu, Y. Lyu, and Z. Liu, "Leap: A lightweight encryption and authentication protocol for in-vehicle communications," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. IEEE, 2019, pp. 1158–1164.

[112] A. Van Herrewege, D. Singelee, and I. Verbauwhede, "Canauth-a simple, backward compatible broadcast authentication protocol for can bus," in *ECRYPT workshop on Lightweight Cryptography*, vol. 2011. ECRYPT, 2011, p. 20.

[113] P. Velan, M. Čermák, P. Čeleda, and M. Drašar, "A survey of methods for encrypted traffic classification and analysis," *International Journal of Network Management*, vol. 25, no. 5, pp. 355–374, 2015.

[114] 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.

[115] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015.

[116] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples," in *International conference on machine learning*. PMLR, 2018, pp. 274–283.

[117] T. Bai, J. Luo, J. Zhao, B. Wen, and Q. Wang, "Recent advances in adversarial training for adversarial robustness," *arXiv preprint arXiv:2102.01356*, 2021.

Acknowledgments

I would like to express my deepest gratitude to my supervisor, Prof. Mauro Conti, for their unwavering support, guidance, and expertise throughout this research. Their invaluable insights and continuous encouragement have greatly contributed to the quality of this thesis. I am also grateful to Prof. Alessandro Brighente, Denis Donadel, Francesco Marchiori, for their feedback, and insightful discussions, which have enriched this work.

The completion of this research would not have been possible without the support and encouragement of my family. I am grateful for their unwavering belief in my abilities and their continuous support throughout my academic journey.

I want to express my heartfelt appreciation to my partner, Nahal, for their unwavering support, understanding, and patience during the challenging phases of this research. Their love and encouragement have been a constant source of motivation and inspiration.

This research would not have been possible without the contributions and support of all these individuals and groups. I am deeply thankful for their involvement and trust in my abilities.