Machine learning for intrusion detection systems in autonomous transportation

Elies Gherbi

université PARIS-SACLAY

Apprentissage automatique pour les systèmes de détection d'intrusion dans les transports autonomes

Machine Learning for intrusion detection systems in autonomous transportation

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°580 d'accréditation, dénomination (STIC)

Spécialité de doctorat : SCIENCE ET TECHNOLOGIES DE L'INFORMATION ET DE LA COMMUNICATION

Unité de recherche : Université Paris-Saclay, Univ Evry, IBISC, 91020, Evry-Courcouronnes, France

Référent : Université d'Evry Val d'Essonne

Thèse présentée et soutenue à Paris-Saclay, Le 05/07/2021, par

Elies GHERBI

Composition du Jury

Anne VILNAT Professeure, Université Paris-Saclay (LISN)	Présidente
Elisa FROMONT Professeure, Université de Rennes (IRISA)	Rapportrice & Examinatrice
Mustapha LEBBAH Maitre de conférences, Université Sorbonne Paris Nord (LIPN)	Rapporteur & Examinateur
Osman SALEM Maitre de conférences, Université Paris Descartes (LIPADE)	Examinateur

Direction de la thèse

Blaise HANCZAR Professeur, Univ.Evry, Université Paris-Saclay (IBISC)	Directeur de thèse
Jean-Christophe JANODET Professeur, Univ.Evry, Université Paris-Saclay (IBISC)	Co-Directeur de thèse
Witold CLAUDEL Docteur, IRT-SystemX	Co-Encadrant, en entreprise

Thèse de doctorat

NNT : 2021UPASG037

Abstract

Despite all the different technological innovations and advances in the automotive field, autonomous vehicles are still in the testing phase. Many actors are working on several improvements in many domains to make autonomous cars the safest option. One of the important dimension is cybersecurity. Autonomous vehicles will be prone to cyberattacks, and criminals might be motivated to hack into the vehicles operating systems and steal essential passenger data or disrupt its operation and jeopardize the passenger's safety. Thus, cybersecurity remains one of the biggest obstacles to overcome to ensure vehicles safety and the contribution that this technology can bring to society.

Indeed, the actual and future design and implementation of Autonomous Vehicles imply many communication interfaces, In-vehicle communication of the embedded system, Vehicle-to-X (V2X) communications between the vehicle and other connected vehicles and structures on the roads. Even though the cybersecurity aspect is incorporated by design, meaning that the system needs to satisfy security standards (anti-virus, firewall, etc.), we can not be sure that all possible breaches are covered. The Intrusion Detection System (IDS) has been introduced in the Information Technology (IT) world to assess the state of the network and detect if a violation occurs. Many experiences and history of IT have inspired the cybersecurity for autonomous vehicles. Nevertheless, autonomous vehicles exhibits their own needs and constraints.

The current state of vehicles evolution has been made possible through successive innovations in many industrial and research fields. Artificial Intelligence (AI) is one of them. It enables learning and implementing most fundamental self-driving tasks. This thesis aims to develop an intelligent in-vehicle Intrusion Detection System (IDS) using Machine Learning (ML) from an automotive perspective, to assess and evaluate the impact of machine learning on enhancing the security of future vehicles. Our primary focus is on In-vehicle communication security. We conduct an empirical investigation to determine the underlying needs and constraints that in-vehicle systems require. First, we review the deep learning literature for anomaly detection and studies on autonomous vehicle intrusion detection system using deep learning. We notice many works on in-vehicle intrusion detection systems, but not all of them consider the constraints of autonomous vehicle systems. In such applications, the data is unbalanced: the rate of normal examples is much higher than the anomalous examples. The emergence of the Generative Adversarial Network (GAN) has recently brought new algorithms for anomaly detection. We develop an adversarial

approach for anomaly detection, based on an Encoding AdversarialNetwork (EAN). Considering the behaviour and the light-weight nature of in-vehicle networks, we show that EAN remains robust to the increase of normal examples modalities, and only a sub-part of the neural network is used for the detection phase.

Controller Area Network (CAN) is one of the most used vehicle bus standard designed to allow microcontrollers and devices to communicate with each others. We propose a Deep CAN intrusion detection system framework. We introduce a Multi-Variate Time Series representation for asynchronous CAN data. We show that this representation enhances the temporal modelling of deep learning architectures for anomaly detection. We study different deep learning tasks (supervised/unsupervised) and compare several architectures to design an in-vehicle intrusion detection system that fits in-vehicle computational constraints.

Future In-vehicle network architecture is composed of different subsystems formed of different Electronic Control Units (ECUs). Each subsystem is responsible for specific services that ensure the autonomous functioning of the vehicle. For functional and security reasons, separate subsystems are isolated, forming a hierarchical architecture of the system. In this thesis, we design a Distributed IDS that fit this in-vehicle architecture system and its constraints and reduces the communication overhead rate induced by the IDS processing.

Résumé

De nombreuse avancés et innovations technologiques sont introduites dans le monde de l'automobile. Plusieurs domaine scientifiques et applicatives contribuent à l'amélioration de ces avancées. L'une des dimensions importantes est la cybersécurité. Effectivement, les véhicules autonomes seront sujets aux cyberattaques et les cybers criminels pourraient pirater les systèmes d'exploitation des véhicules et perturber leur fonctionnement et mettre en danger la sureté des passagers. Ainsi, la cybersécurité reste un obstacle à surmonter pour sécuriser les véhicules et permettre aux innovations technologiques dans le domaine des transports d'apporter des solutions aux problèmes de la société et éviter leur détournement à des fin malicieuses. En effet, la conception actuelles et futures des véhicules autonomes implique de nombreuses interfaces de communication, la communication dans le véhicule entre les différents systèmes embarqués, les communications Vehicle-to-X (V2X) entre le véhicule et d'autres véhicules et structures connectés sur les routes. Plusieurs mécanismes de défense sont implémentés pour répondre aux normes de sécurité (antivirus, pare-feu, etc.), mais nous ne pouvons pas être sûrs que toutes les failles possibles sont couvertes, spécialement dans des systèmes complexes comme les voitures autonomes. Le système de détection d'intrusion a été introduit dans le monde Information Technology (IT) pour évaluer l'état du réseau et détecter tous comportement malveillant. Le mon de l'IT a connus beaucoup plus d'expérience en termes de mécanisme de défense qui peut inspirer la cybersécurité des transports intelligent (voiture autonomes), néanmoins, ces dernières requirent leurs propres besoins et contraintes lié à la sureté et aussi à leur architecture system. L'état actuel de l'évolution des véhicules a été rendu possible grâce à des innovations successives dans de nombreux domaines industriels et de recherche. L'intelligence artificielle en fait partie, ces diffèrent technique permettent d'apprendre et de mettre en œuvre des tâches complexe tel que la conduite autonome. Cette thèse vise à développer un system intelligent de détection d'intrusion en utilisant l'apprentissage automatique dans un contexte automobile. L'objective est d'évaluer l'impact de l'apprentissage automatique sur l'amélioration de la sécurité des véhicules futurs (autonomes). Notre objectif principal est la sécurité des communications entres les différents systèmes dans la voiture. Dans ce but, nous menons une enquête empirique pour déterminer les besoins sous-jacents et les contraintes qu'exigent les systèmes embarqués. Nous passons en revue la littérature d'apprentissage profond pour la détection d'anomalie, on note qu'il y a un manque d'étude personnalisée sur le système de détection d'intrusion de véhicule autonome utilisant l'apprentissage profond. Dans de telles applications, les données sont

déséquilibrées : le taux d'exemples normaux est beaucoup plus élevé que les exemples anormaux. L'émergence du Generative Adversarial Network (GAN) a récemment apporté de nouveaux algorithmes pour la détection des anomalies. Nous développons une approche antagoniste (adversarial) pour la détection des anomalies, basée sur un Encoding AdversarialNetwork (EAN). Compte tenu du comportement et de la légèreté des réseaux embarqués, nous montrons que EAN reste robuste à l'augmentation des modalités d'exemples normaux, et seule une sous-partie du réseau neuronal est utilisée pour la phase de détection.

Controller Area Network (CAN) est l'une des normes de bus de données très répandu dans les véhicules, conçue pour permettre aux microcontrôleurs de communiquer entre eux. Nous proposons un système de détection d'intrusion Deep CAN. Nous introduisons une représentation de séries temporelles à variables multiples pour les données asynchrones CAN. Nous montrons que cette représentation améliore la modélisation temporelle des architectures d'apprentissage profond pour la détection d'anomalies. Nous étudions différentes tâches d'apprentissage profond (supervisées / non supervisées) et comparons plusieurs architectures pour concevoir un système de détection d'intrusion embarqué qui s'adapte aux contraintes de calcul des systèmes faible en ressource. L'architecture future des réseaux embarqués dans les véhicules sont composées de différents sous-systèmes. Chaque sous-système est responsable de services spécifiques qui assurent le fonctionnement autonome du véhicule. Pour des raisons fonctionnelles et de sécurité, les sous-systèmes sont isolés, formant une architecture de communication hiérarchique de l'ensemble du système. Dans cette thèse, nous concevons un IDS distribué qui s'adapte à l'architecture embarquée et à ses contraintes et réduit le taux de surcharge de communication induit par le traitement de l'IDS.

Contents

List of Figures

List of Tables

Acronyms

AI Artificial Intelligence. i, ix, 10

CAN Controller Area Network. ii, iv, 26

DDoS Distributed Denial of Service attack. 33

DoS Denial Of Service. 31, 33

EAN Encoding AdversarialNetwork. ii, iv

ECU Electronic Control Unit. 24, 26, 38

ECUs Electronic Control Units. ii, 24, 25, 38, 40, 42

GAN Generative Adversarial Network. i, iv

IDS Intrusion Detection System. i, ii, iv, 21, 44

IOT Internet Of Things. 3, 23

IT Information Technology. i, iii, 3, 7, 11, 12, 15, 21, 36

ITS Intelligent Transportation System. 1–5, 12–14, 21, 22, 34, 36

ITSs Intelligent Transportation Systems. 2, 3

MITM Man-In-The-Middle. 31

ML Machine Learning. i

R2L Remote to Local. 33

U2R User to Root. 33

V2I vehicle-to-infrastructure. 23

V2V Vehicle-to-Vehicle. 23

V2X Vehicle-to-X. 21, 40

VANETs Vehicular Adhoc Networks. 23, 36, 41

Chapter 1

Introduction

1.1 Intelligent Transportation Systems (ITSs)

A long time ago, human beings did not have any means of transportation; they were used to travel only on their foot and carrying their goods either on their backs or using animals. Until the invention of the wheel, it was a starting of a long series of transportation innovation. We overcame natural obstacles like water and seas; we replaced animal power in 1776 when James Watt invented the steam engine. This innovation was a revolution in transportations, which brought a lot to the evolution of today's society. From empowering the ships and locomotives to increase the efficiency of the human operations to achieve the goal of flying and reach the moon. We never stopped since then, we have come across a lot of innovation in transportation, and today's society is more dependent on its transportation systems, which represents a vital element for the proper and safe functioning of society. The transportations systems have always been deeply affected by major technologies shifts. These last years, a lot of efforts have been made to make the transportation more intelligent and enabling them to offer different services through applications, ranging from automation and driver assistance to infotainment applications. The growth of cities and the rise of the urban population brought challenges like congestion, air pollution, and road accidents. To deal with these issues, full automation for transportations systems arises ITS (intelligent transportation system). The advance in Artificial Intelligence (AI), more precisely machine learning by using deep learning methods, has drawn a lot of attention in both scientific and industrial actors. It enables many functionalities that rely on accurate decision making, from trying to learn traffic rules to coping with human pedestrian unpredictability in many different contexts (autonomous vehicles, drones and autonomous train). ITS aims to reach a certain level of autonomy that allows the scalability of mobility more smartly and optimally. Indeed, if all the cars are driverless and connected, everything is predictable. The reaction time is at its minimum and guarantees a well-distributed and resource-efficient system, which means lowering the numbers of vehicles. Some study estimate Autonomous Vehicle can reduce the number of accident by 40% Fagnant und Kockelman (2015). The current state of the motivation is clear, ITS will save lives and enables an

optimal and safer ecosystem for the future cities, if and only if well defended, especially on cybersecurity threats. The technology has the decision making control, if the ITS are prone to cyberattack, the results can be disastrous.

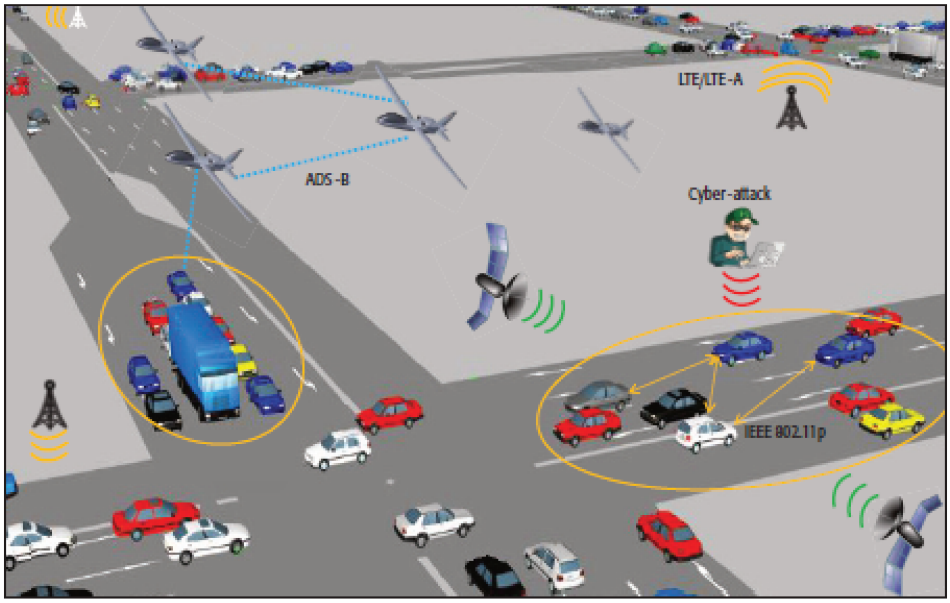


Figure 1.1: Schema describing the synopsis of Intelligent transportation systems.Sedjelmaci u. a. (2019)

1.2 Cybersecurity in Intelligent Transportation Systems

The Intelligent Transportation Systems (ITSs) could be subject to a variety of cyberattacks. If a physical device connects to the Internet, it exposes the vehicle to intrusions attempts via its connection interfaces. Therefore, the implementation and the design must survive attacks if an intrusion occurs. Attacking the transportation infrastructure can now be accomplished by attacking these ITSs information systems in the same manner that computer hackers attempt to break into other information systems Sedjelmaci u. a. (2019).

For example, in railways, classified as critical infrastructures, If something goes wrong in signalling technology, it may have massive consequences: (i) Trains stop (emergency braking, system failures), (ii) Negative economic effects and loss of trust, (iii) in the worst case, accident casualties. Recently, a couple of cyberattacks target rail transport. In 2008, a teenager wreaked havoc by derailing four tram-trains in Lodz, Poland using an adapted TV remote. There were several injuries. In 2011, hackers remotely attacked computers in the north-western USA, stopping railway signals for two days. In 2015, North Korea was suspected of hacking a subway operator in Seoul, South Korea, over several months. Dozens of terminals were infected with malware.

Drones, parts of the ITSs, also represent an attractive target for attackers. Recently, an Iranian hacker used a jamming attack to interrupt the communication signals between a controller and a drone so that the drone switches

to autopilot mode. The latter relied on GPS coordinates to guide itself back to its home base. Afterwards, the hacker launched a GPS spoofing attack to falsify GPS coordinates and led the drone to think it was close to the home base. Furthermore, with vehicle composed of numerous intelligence components that control the different functions, the massive introduction of electronics units requests a rigorous approach to the overall security of those parts.

Besides, the emergence of the connected vehicle concept, threats to the safety and privacy of motorists and passengers already exist and are set to grow substantially. At the Black Hat security conference, automotive cybersecurity researchers presented a new arsenal of attacks against the Jeep Cherokee through WIFI connection Charlie Miller (2015). These new threats require the setup of new architectures, access control mechanisms and particularly monitoring facilities to detect suspicious behaviours (potentially intrusions) and if necessary to take defensive actions to eliminate or limit the impact of these cyberattacks in the origin of these misbehaviours.

ITSs are complex, time-critical systems in which the physical safety of road users and the efficiency of the transportation services directly depend on the provision of cybersecurity. A comprehensive standard and the creation of a security strategy are not yet a fact. Some of the security technologies are in the initial stage of research regarding their application in ITSs. We can summarize that ITSs innovation is an application of information and communication technologies to the transportation domain. As such, ITS reuse existing technologies to create innovative services that can be applied in every transportation mode used by passenger or freight service. ITSs architectures and challenges are different from the application of information and technologies to the IT word. Indeed, they do not meet the same standard and safety requirement, especially in terms of resources (memory, computation and communication), and they do not operate in the same context.

An ITS can be framed as a part of Internet Of Things (IOT) and so it can be outlined using the same architecture Fig 1.2. In the ITS perspective, it gives each layer a more propoer functions.

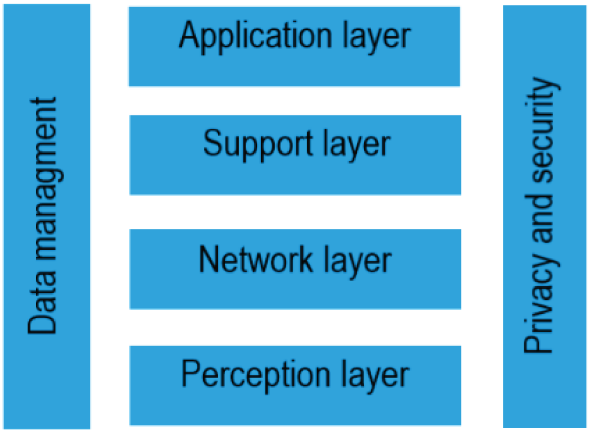


Figure 1.2: IoT architecture outlines

- **Perception layer:** It encompasses all the sensors that can gather information, even the infrastructure devices.

It is more related to the internal vehicular network design, and many security issues are concerned.

- **Network layer:** In the ITS systems, the network layer is a complex mesh of wired and wireless technologies. One of this layer's main security objective and challenges is to provide node authentification in VANET with constraints, such as protecting personnel data and authentification. To this end, many technologies and standards are proposed. V2X has been widely explored to fulfil the requirement for duplication protection, integrity, confidentiality Mandy und Mahgoub (2018).

- **Support layer:** With the emergence of technologies like Fog or Cloud computing, the data is processed based on their time-sensitive function and spatial need consideration Khan u. a. (2017). So, challenges like protecting operations distributed in the Fog systems appear, while the existing security and privacy measurement designed for cloud systems cannot be directly applied.

- **Application layer:** It's the interface of the interaction with the final user (mainly in ITS it could be the passenger). The interaction can be expressed differently, depending on the final aim of the specific application. The data acquired in the perception layer can be analyzed and processed in multiple ways (local, cloud), depending on the data semantics and its time constraint. Therefore, the security actions toward those applications need to feet the time constraint calculation and the topology of the data flow in the vehicle. Note that a major attack aims at the layer because it can interact with the passenger if blackmail is the goal.

We define two separate cybersecurity concerns for ITS, In-vehicle cybersecurity and external vehicle cybersecurity. Both need to be considered at the implementation of the security layer for ITS. In this thesis, our work's primary focus is on the in-vehicle network system (autonomous cars)(See Fig 1.3). The external cybersecurity (communication between the vehicles and things) is not taken into account in the scope of this thesis.

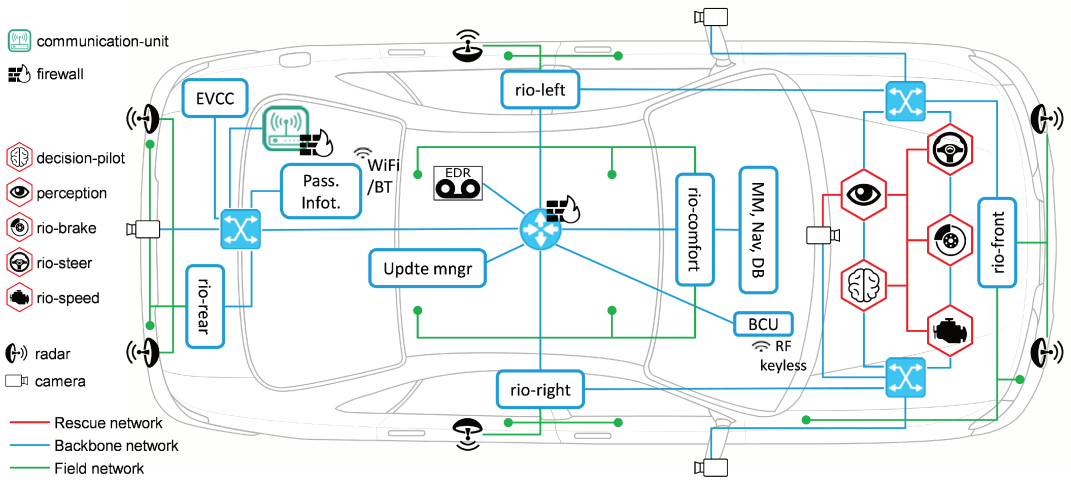


Figure 1.3: In-vehicle network architecture of autonomous vehicles

1.3 In-vehicle cybersecurity

Although ITS are relatively new, especially autonomous vehicles (automotive transportation), many research and industrial areas have integrated and tested cybersecurity methods by conducting many experiments on conventional ITS. It is ranging from strong authentification, regular auditing, encrypted communication to a private network and secure routing Qiang Hu (2018); Hahn u. a. (2019); Jadoon u. a. (2018). The conventional methods, like Cryptographic approaches, are not entirely suitable for ITS. They need to meet a set of requirement like low latency, lightweight encryption and communication overhead. We note that the ITS architecture can differ in terms of implementation details depending on the constructors, where nowadays, many actors are working on in-vehicle architectures systems that are reliable in terms of cybersecurity. These architectures and their implementations are still in their prototyping phase (See Fig 1.3).

In autonomous vehicles, the external interfaces gradually increase, as well as the cybersecurity threats. For example, in Charlie Miller (2015) the authors achieved the invasion of Toyota Prius and Ford Escape and remotely invaded a Jeep Cherokee. Furthermore, in Smith (2015) the authors demonstrated possible paths of vehicle cyber attacking using penetration test. Researchers analyze the security issues in vehicles by experimental attacking Checkoway u. a. (2011); Koscher u. a. (2010a), the results show the severity of vehicular cyberattacks. In Petit und Shladover (2014), the authors investigates potential network attacks and vulnerabilities for autonomous vehicles, the possible attack targets include traffic signs, machine vision, GPS signals, sensors, radar signals, lidar signals, navigation). Attacks against vehicles can be divided into logical attacks (password attacks, software attacks, communications attacks, etc.), physical attacks (side channel attacks, denial of service attacks, interference attacks, penetration attacks, tamper attacks, etc.) and other attacks Wolf (2009).

The embedded cyber defence architecture of automotive transportation aims to protect the information transfer, sensor signals and critical passengers data by monitoring the communications among ECUs, sensors, and gateways in the in-vehicle networks by including message authentication, data encryption, and intrusion protection.

1.3.1 In-vehicle network communication system

It represents an extensive system where many computers unsure functionalities by interacting and communicating with each other, using different automotive field busses like CAN (Controller Area Network), FlexRay, LIN(Local Interconnect Network). Mainly, those automotive systems lack security, and they handle many features such as real-time communication between controllers, engine management. The Intrusion Detection System primary role is to monitor those systems and detect malicious attempts or attacks and alert the system's user.

In-vehicles communications are mainly signals and data transfer carried by automotive network protocols; some typical networks protocols are introduced in the following:

- **Controller Area Network (CAN):** CAN bus is the most widely used bus field in the automotive industry. It plays a significant role in the automotive power system and comfort system. The maximum speed of a high-speed CAN bus can reach 1 Mbps. The data field length of the CAN frame is 8 bytes. The cyclic redundancy check (CRC) is used to ensure the correctness of transmission. (In Chapter 2, we will introduce a detailed functional description of CAN.)

- **Local Interconnect Network (LIN):** LIN bus is mainly used for controlling the vehicles seats, doors, wipers, sunroof, and so on. For LIN bus, the maximum of the data field in a frame is 8 bytes. The checksum is calculated to verify that the integrity of the message has been preserved during transmission operation. Unlike CAN bus, LIN bus uses the master/slave node mode for communication.

- **FlexRay:** FlexRay bus has a higher transferring speed than CAN bus and mainly used the automotive power control system. The maximum speed of FlexRay can be up to 10 Mbps. Time-division multiple access (TDMA) and flexible time division multiple access (FTDMA) are used in FlexRay to ensure the real-time requirements of network communication. The data field length of FlexRay frame is 254 bytes. The CRC is used to check errors during bus communication. This protocol differs from CAN by its performance in terms of throughput and higher reliability. The cost of a FlexRay node is currently higher than that of a CAN node, which is programmed to be the de facto replacement in automotive electrical and electronic architectures.

- **Ethernet:** The Ethernet plays a significant role in the new automotive architecture. The IEEE 802.1 AVB (Garner et al., 2007) and TTEthernet (AS6802) (SAE, 2016b) based on the automotive Ethernet have been used for infotainment application. The Ethernet is also the foundation of the Internet protocol. With the Internet protocol, remote calibration, remote diagnosis and remote update between vehicle and server can be achieved. For the automotive Ethernet, transmission speed can be 100/1000 Mbps, the data field length of Ethernet frame can be more than a thousand bytes. The CRC is used in Ethernet frame to check to ensure the accuracy of data transmission Matheus und Königseder (2015).

1.3.2 Basic in-vehicle security network architecture

Compared with traditional vehicles, connected vehicles require more information transfer. Sensor signals and critical data must be protected to ensure the safety of connected vehicles Qiang Hu (2018). In-vehicle networks connect the communications among ECUs, sensors, and gateways. The design of each layer must take security into consideration as in the following:

- **Individual ECU Layer:** This layer contains software trusted execution and data protection provides hardware foundation for upper layers security mechanism.

- **In-vehicle Network Layer:** Cryptography related mechanisms can be used to encrypt the transferring data in the network.
- **Gateway Layer:** Gateway layer has the critical security functions, including access control and intrusion detection, this layer helps the data exchange in different network domain Seifert und Obermaisser (2014).
- **Firewall Layer:** Firewall layer is used in general to protect the outside communication interfaces of vehicles, such as OBD-II interface, V2X on-board unit, and infotainment system (it also can be implemented for the in-vehicle network communication based on the architecture design and isolation)Luo und Hou (2019).
- **Intrusion Detection:** Intrusion detection layer is used to monitor all the network communication in the in-vehicle system and detect any misbehaviour or anomalies in the network. (The misbehaviour could be a result of an attack or an intrusion attempt).

The technological advance enables automated driving features and many others like infotainment. But, it brings a substantial rise in functional complexity regarding utilized algorithms—also, the number of processed information increases over several Electronic Control Units (ECUs). The increase of complex features is challenging to handle with the existing architectures. The introduction of many hierarchy levels in the logical system facilitates functional elements with wide internal variety and distribution over several subsystems. In-vehicle IDS need to meet the constraints of that hierarchical architecture. Also, the resources induced by the IDS are limited in the embedded systems. Today, most attacks targeting IT are well represented in the literature, but cybersecurity in an embedded system is relatively new, especially for future Autonomous Vehicle.

The expansive and broad definition of AI and ML can and should be applied in cybersecurity, encompassing various methods that have developed over many decades have demonstrated effectiveness in many other application domains. AI/ML is viewed as a necessary response to the continuing growth in the number and complexity of threats, the evolving nature of threats, and the need for rapid (and therefore substantially automatic) responses to detected threats. We discuss this point in the following section.

1.4 Artificial intelligence for Cybersecurity

1.4.1 Artificial intelligence

Since the rise of homo sapiens, millions of years ago, our ancestors always achieved abstract thinking, communicated complex thought, and accumulated and transmits information. That led our primary communities into a continues evolution through time and their surrounding environment better than any other species. Biologically compared to apes, we have some crucial and minor changes in brain size and some neurological organisations that somehow enhances our cognitive ability. So, as a human on this planet, we name this distinguishing ability Intelligence. We always used this ability, so with science development in its different fields, it is logical to understand this intelligence to develop efficient, productive technologies Bostrom (2014); McCorduck (2004). With the advanced research in neuroscience and biological systems, with continuous growth after both agricultural and industrial revolution, especially after the invention of computers in the 1940s, we have been expecting machine matching humans in general intelligence by possessing a common sense and an effective ability to learn. We plan to meet complex information-processing challenges across a wide range of natural and abstract domains.Bostrom (2014). In the following Fig 1.6, we highlight some significant events that led us to Artificial Intelligence nowadays.

Taking a look at the succession of events in Fig 1.6, Th AI thinking mindset goes from making machines as good as human or even much better, from where the hypothesis of "The singularity " arises:

"Let an ultraintelligent machine be defined as a machine that can far surpass all the intellectual activities of any man however clever. Since the design of machines is one of these intellectual activities, an ultraintelligent machine could design even better machines; there would then unquestionably be an 'intelligence explosion,' and the intelligence of man would be left far behind. Thus the first ultraintelligent machine is the last invention that man need ever make."

— Irving John Good (1966)

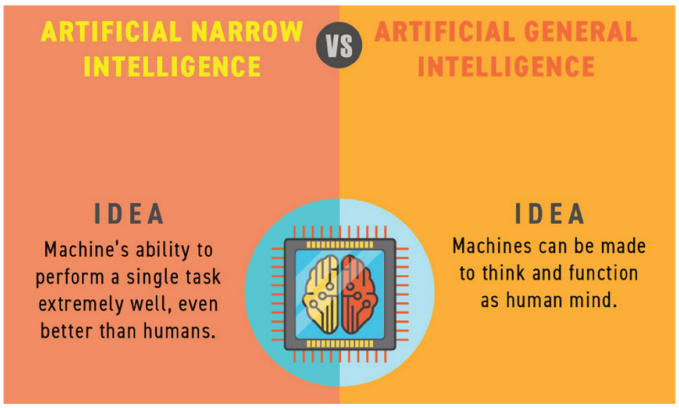


Figure 1.4: Narrow AI vs General AI - Credit India Analytics Magazine

Next, it had some downfall "AI winter", the AI thinking mindest get less ambitious, the ***expert system*** had success, and brought another wave of interest to the domain. These ***expert system*** focused on much narrower tasks. They were rule-based programs that make an inference on the knowledge defined by experts and after hand-coded in a formal language. Those approaches were expensive, and needed continues update. In general, they were not practical Bostrom (2014). Then, a newly popular technique grouped under the umbrella of *Machine learning*(ML) including neural network (NN) and genetic algorithm promised to overcome some of the shortcomings of the *Good Old Fashion Artificial intelligence* (GOFAI). Researchers had shown that a neural network could learn from experience, finding ways of generalizing from examples and discovering hidden statistical patterns in their data input Fukushima (1980); Lecun u. a. (1998); Hinton und Salakhutdinov (2006). They demonstrated that NN could accurately tackle the classification task and pattern recognition problems without an explicit definition of the features extracted or manual weighting thanks to the backpropagation algorithm. NN along with other algorithms like *K-nearest-neighbors, support vector machines (SVM), naive bayes* among others have been successful; One of the reason is their ability to learn hight level abstraction concepts given raw data input data. (Figure classical ML based on handcrafted features VS NN workflow). In general, AI goes from Artificial General Intelligence to Artificial narrow Intelligence(See Fig 1.4).

The Fig 1.5 shows machine learning segmentation per their learning paradigm and their eventual applications.

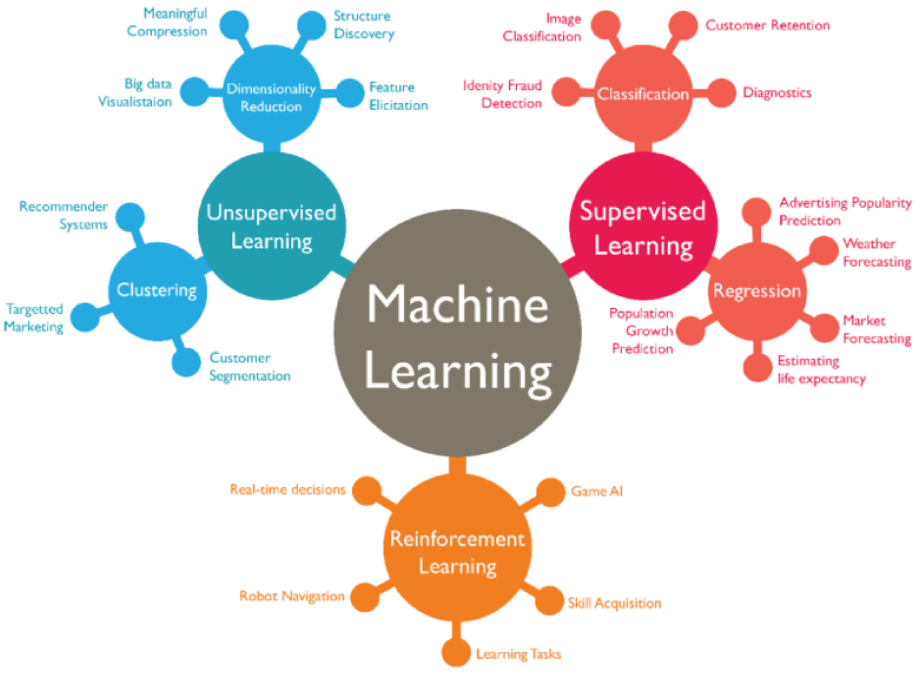


Figure 1.5: Machines Learinng Segmentation - Credit unknown, traced to A. Wahid (gleetech) Alseg

Broadly speaking, Ml is a field of study that gives computers the ability to learn without being explicitly programmed **[Arthur Samuel, 1959]**. ML has achieved a rapid growth and has drawn lots of attention thanks to recent advances of Deep Learning (DL) which is a sub-filed of ML based on Artificial Neural Network (ANNs).

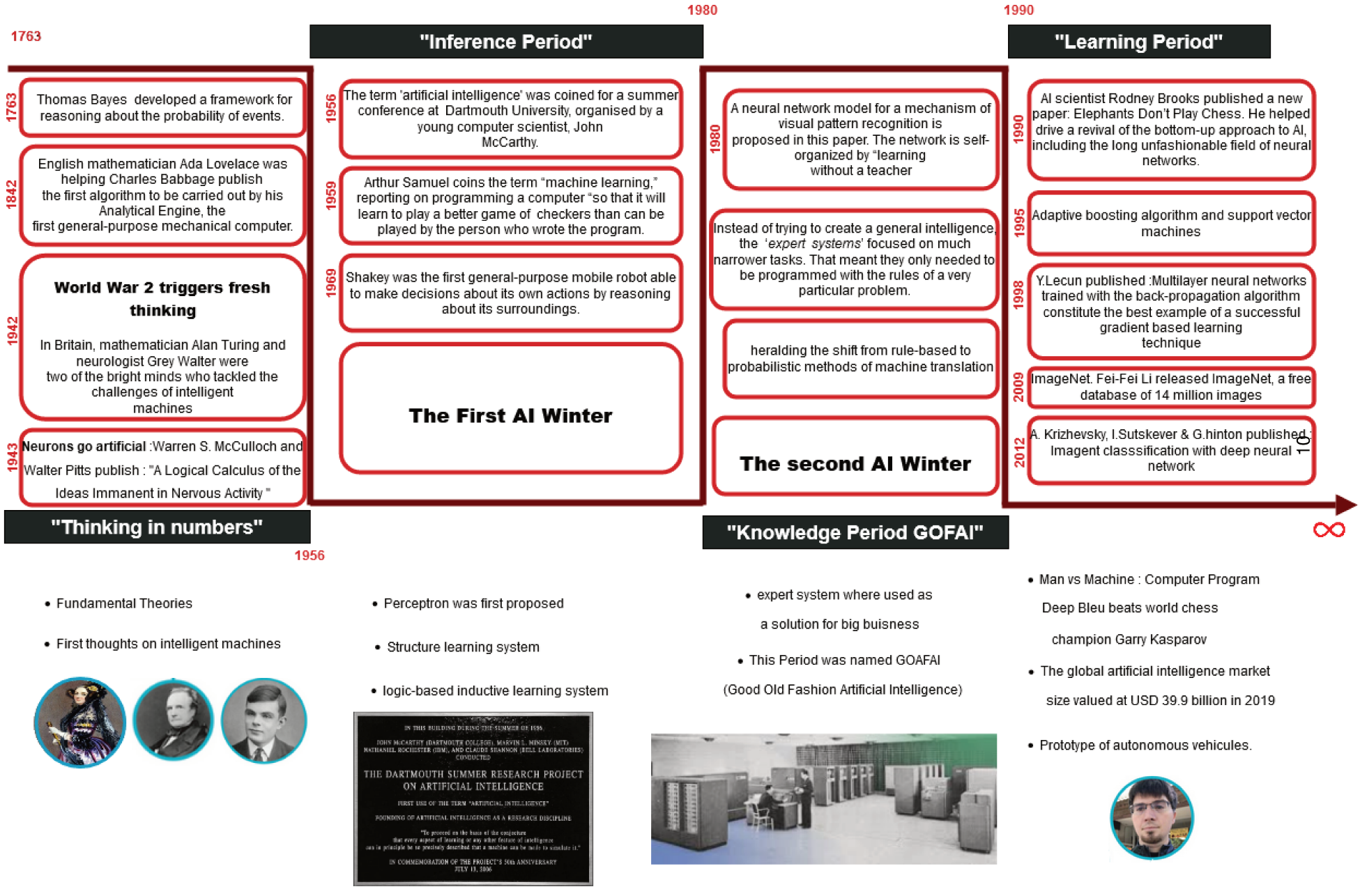


Figure 1.6: Historical developement of AI Bayes (1763); Mcculloch und Pitts (1943); Fukushima (1980); Brooks (1990); Cortes und Vapnik (1995); Bostrom (2014); Lecun u. a (1998)

1.4.2 Machine learning utility for Cybersecurity

Cybersecurity is a very broad term referring to everything related to the protection of cyber resources. As much as any other domain, cybersecurity had a lot of interest in using ML approaches. It's considered as a necessary response to face the continuous growth of the number of threats, and the need for rapid, substantially automatic responses to detected threats. The primary targets for ML application in cybersecurity are Intrusion Detection Systems that we will review in the chapter 2, characterization (malicious code), user behavioural modelling, automated vulnerability testing and intrusion defence. Indeed, ML is beneficial to cut through the large volume of data and find indicators of compromisation and unwanted behaviour using correlations across data sources. These systems would assist human analysts by elevating or alerting them to significant events that require responses without overwhelming the organization with false alarms or other unnecessary indicators Loaiza u. a. (2019).

However, in general, ML has had promising results in improving the efficacy of cybersecurity technologies such as endpoint security to detect and prevent new and previously unseen malware. Machine learning intends to be incorporated into seemingly every new cybersecurity defence applications to avoid any compromisation since attackers are evolving and adapting at the same scale and at a faster pace than defenders. The primary benefits anticipated are:

- Improvements to the investigation of security alerts.
- Improvements in accuracy and reduced false-positive rates.
- The elimination of more compute-intensive detection techniques.
- detection of zero-day threats.

The most practical applications reside in the Information Technology word (IT) where the most data and experience coping with cybersecurity threats and attacks are available. It's recent that data related to network information and cybersecurity are starting to be leveraged for public research and analysis in competition platforms like **Kaggel**. But, mainly, companies that have a requirement to train machine learning algorithms, especially DL algorithms that are data-hungry are those who benefit more from DL technologies in their private research. In fact, more than half of the respondents report they are using machine learning technology for cybersecurity purposes to some degree, up from 47% in 2018, per the 29% of those companies leveraging machine learning extensively. This level of adoption has made machine learning a foundational cybersecurity technology and mostly applicable for specific use cases oraclAIcs.

The lack of datasets for research in this area is a problem. Some vendors have large volumes of data available (for example, network switch/infrastructure vendors, and providers of anti-virus and network/computer monitor software and systems). However, the most studied available datasets are dated (DARPA 1998 and 1999, and KDD 1999 data), and the characteristics and volume of attacks have significantly changed since that time.

1.5 Machine learning for a future in-vehicle intrusion detection system

Intrusion detection systems (IDSs) are common to defend against Cyber-Physical system (CPS) attacks, especially the Controller Area Network vulnerabilities. These systems monitor the ECUs networks such as CAN and report malicious activities. In the CPS domain, an IDS can detect attackers attempting to modify or misrepresent physical processes. In other words, if an attacker intends to cause the driver to speed, he may choose to inject packets detailing a lower speed, which would cause the speedometer to display incorrect information. In this case, an effective IDS will notice that the data for speed does not conform to the expected behaviour indicated by the data for the related physical processes (e.g., engine and wheel rotational velocities, fuel consumption). So, the IDS will notice that the speed readings are anomalous. As explained, the example seems simple, and this case can be hand-coded in a set of rules to prevent this specific attack. But as soon as we start enumerating the number of information that can be changed while corrupting the in-vehicle system, it leads to an explosion and intractable scenarios. As another example, if an IDS knows that a substantial increase in an automobile's brake pressure likely precedes a relative decrease in velocity, the IDS can assert that no change, a small change, or an increase in velocity (after significant brake pressure) is abnormal. Of course, this requires an IDS capable of determining expected behaviour and identifying anomalies.

In a nutshell, to design such an IDS for a vulnerable CPS, embryonic IDS architects require the following:

- An ample quantity of knowledge under normal operating conditions to establish normal behaviour.
- Capture the dynamics or patterns of a CPS, to include an understanding of the current system state or behaviour enables predictions concerning a future state.
- Solid understanding of a system dynamic, how one signal affects another AND how the historical state defines the current state (causality).
- A process to determine whether new traffic conforms to normal behaviour.
- An alert system to report to the administrator the traffic that does not conform.

In-vehicle networks are traditionally simpler than IT networks, yet, CAN is crucial to ensure message transmission by eliminating conflicts and be resilient to noise. As the number of electronic components penetrating vehicle subsystems increases, the in-vehicle networks are given significance as a medium for exchanging information. As mentioned above in the Sub-section 1.3.2, the distributed functions among different subsystems triggered the complexity level of networks and transformed in-vehicle networks into advanced controller area networks. Upcoming autonomous driving abilities and ITS applications that require the collaboration of several subsystems force automotive manufacturers and equipment suppliers to add new hardware and software layers to their products, making

in-vehicle networks a source of data. By increasing connectivity surfaces of vehicles to support ITS, vehicles are becoming vulnerable against malice and misbehaviours.

In that context, we have two primary assumptions. i) The system behaviour is well defined; it uses deterministic communication between distinct components. ii) The system can behave differently depending on various context (for example, monitoring a car behaviour can change following the area and the state of the road). It is tough to develop an IDS by explicitly coding all rules for each specific context. After the aforementioned ML success and their ability to learn specific tasks without being explicitly programmed, the straightforward application of ML methods for In-vehicle IDS system is well expected (we review those methods in the Chapter3). Deep learning has been used successfully in many domains and task, and cybersecurity also got inspired by machine learning in general to develop an intelligent Intrusion Detection System. To have an intelligent in-vehicle IDS, we need to take into consideration the following constraints:

- Modeling the normal behaviour of a vehicle following different context.
- In the automotive context, many cyber physical systems are computationally limited by available hardware or by standards and regulations.
- Ensure high accuracy and reduce false alarm rate.
- A distributed intrusion detection system should fit the actual network and system design.
- Monitoring of the system and real-time detection.
- Detection process with low resource consumption.
- Reduce the communication overhead induced by the integration of IDS in the In-vehicle network.

However, systems that incorporate advanced algorithms such as AI and ML must be properly designed to accept and process the high arrival rates of network and measurement data, as must the software-based sensors (Probes) that collect data from ECUs attached to the network, to ensure that the performance network is not unduly compromised.

1.6 Contribution and Outline

As Introduced above, this thesis mainly came across three significant domains, Intelligent transportation systems (ITS) represent our study's context, Artificial intelligence as the central technology used and explored to solve Cybersecurity problem and challenges related to the ITS application. So we can position precisely the scope and the purpose of our contribution in this thesis in the following.

1.6.1 Research question and constraints

As the research presented in this thesis developed above, it naturally formed four main parts: Intelligent intrusion detection system STATE-OF-THE-ART, Adversarial Learning for Anomaly Detection, Empirical Time series evaluation of In-vehicle Intrusion detection system, Distributed anomaly detection based In-vehicle IDS. Specific aims and objectives related to each of these parts of the thesis are presents in their respective chapters, whilst general research questions are presented here at a high level.

- How machine learning can enhance the existing IDS systems?
- What are the constraints, requirements fixed by the ITS domains to assess the utility of ML in in-vehicle context?
- What are the most suitable ML approaches that fit the IDS objective while respecting the requirement and constraint?
- How to deal with the lack of data to validate the proposal?
- How to build a ML model tailored for the in-vehicle system?
- If data available, how to cope with an imbalanced rate between normal examples and the lack of attacks examples?
- The technologies emerging for the in-vehicle network are new and still evolving, how to avoid the obsolescence?
- Major of data sets in this domain are not in clear (encoded or encrypted), how the model can use those data?
- How the model takes into consideration the topology of this data?
- How we can integrate the model by respecting the logical architecture of future in-vehicle systems?

This thesis's scope is determined by several pragmatic conditions, which have been applied to ensure a focused investigation without compromising the ability to answer the research questions and respond to the project's finding expectation that embraces this research. It is necessary since the thesis considers several large research domains and works as part of an R&D project. First, this thesis's empirical work only considers network-based anomaly detection as the main path of contribution. However, the review of the domain considers all the main methods of intrusion detection. Although some challenges and concepts apply to other domain applications like healthcare and fraud detection, these applications are excluded as an application, but the related theoretical work has been considered. Furthermore, since this investigation focuses on machine learning, other, conventional techniques

applied to intrusion detection are not considered. However, the review does consider a broad range of different AI techniques applied to intrusion detection.

There are several aspects of intrusion detection that are not considered here, although they would require attention when developing an IDS to be deployed in real life, such as:

Architecture: the focus here is on what could be referred to as a detection module that would exist in a larger IDS framework that takes in consideration many others method (Hybrid). Especially in-vehicle networks, the architecture is very important. This includes determining where to deploy the IDS and considering the local information, and it's safety impact, which is considered a general challenge.

Data collection: since the KDD data set (is more IT and old dataset) is adopted in this work to benchmark the ML methods, data collection is required. However, it would be necessary to collect data from the environment in which an IDS is to be employed (real autonomous car in real life). In our case, the data collection has been done to better understand the environment of the in-vehicle network and analyse the data topology and characteristics (This also includes labelling data for supervised learning.).

Data preprocessing: some data preprocessing is necessary for this work, adopted, enumerating and scaling feature values. However, In this domain, it was a lack of specific analysis of CAN data, so a convenient and custom Data preprocessing is required to fit the ML inputs.

Performance: there are several mechanisms that can be adopted to help achieve a better performing IDS, in terms of detection rates, speed and memory usage, and also communication overhead. This thesis focuses on the issues and challenges posed by the research questions and helps develop an optimal and practical in-vehicle IDS prototype.

Other pragmatic considerations: detecting new intrusions will always be a challenge; there will always be new software, which inevitably has vulnerabilities that can be exploited. Therefore, re-training is necessary once new data is available. When and how this is done is considered here by adopting a general framework that combines supervised learning and unsupervised learning.

1.6.2 Research Methodology and routing

There are three main contributions to this thesis. These parts have made contributions to both the intrusion detection for in-vehicle systems and machine learning domains. However, the focus of this thesis is on applying machine learning to the in-vehicle intrusion detection system. We propose practical solutions to tackle some challenges presented above with a customization effort to developpe tailored deep learning architectures that respond to the constraint of the future in-vehicle network. This study starts with an extensive literature review of the following areas:

- Intrusion detection system.
- Machine learning for anomaly detection with a focus on deep adversarial learning.

- In-vehicle CAN data analysis.
- Supervised and Unsupervised DL for time series analysis
- Distributed DL architectures

We present the flow of the thesis work as follows:

In the Part I (Chapter 2 and Chapter 3) , we provide the literature survey on the existing state-of-the-art methods, taxonomy related to the area mentioned above. It inspires ideas and hypotheses to build the problematics that guide our research. Each problem is defined in their respective contribution chapter.

In the Chapter 4, we propose our first contribution, based on the SOA analysis, we start with a contribution to the anomaly detection area. Anomaly detection is a standard problem in Machine Learning with various applications such as health-care, predictive maintenance, and cyber-security. In such applications, the data is unbalanced: the rate of regular examples is much higher than the anomalous examples. The emergence of the Generative Adversarial Networks (GANs) has recently brought new algorithms for anomaly detection. Most of them use the generator as a proxy for the reconstruction loss. The idea is that the generator cannot reconstruct an anomaly. We develop an alternative approach for anomaly detection, based on an Encoding Adversarial Network (AnoEAN), which maps the data to a latent space (decision space), where the detection of anomalies is done directly by calculating a score. Our encoder is learned by adversarial learning, using two loss functions, the first constraining the encoder to project regular data into a Gaussian distribution and the second, to project anomalous data outside this distribution. We conduct a series of experiments on several standard bases and show that our approach outperforms the state of the art when using 10% anomalies during the learning stage, and detects unseen anomalies.

In the Chapter 5, In this work, we propose a Deep CAN intrusion detection system framework. We introduce a multivariate time series representation for asynchronous CAN data, enhancing the temporal modelling of deep learning architectures for anomaly detection. We study different deep learning tasks (supervised/unsupervised) and compare several architectures to design an in-vehicle intrusion detection system that fits in-vehicle computational constraints. Our system is time window wise: any given time frame is labelled either anomalous or normal. We conduct experiments with many types of attacks on an in-vehicle CAN using SynCAn Dataset. We show that our system yields good results and allows us to detect many kinds of attacks.

In Chapter 6 : In this contribution, the in-vehicle network architecture is hierarchically distributed where the different sub-network are monitored with various and independent probes. We propose a Distributed Anomaly detection IDS (DAD) that fits the distributed in-vehicle architecture. To this end, we use multi-modal deep learning architecture. The model captures each probe view pattern as a feature vector representation of the input through a

sequence modelling transformation using a Temporal Convolutional Neural Network (TCN). Our proposed method brings two optimisations compared to central models. First, it reduces the transmitted vector's size between the probes and the central probe (Bastion). Second, we introduced a new flag e_i related to each probe that lower the computational inference time and the intrusion detection system's communication overhead.

1.6.3 Publications

- Elies Gherbi, Blaise Hanczar, Jean-Christophe Janodet, Witold Klaudel. Construction d'espace latentpour la détection d'anomalies par apprentissage adversarial. Conférence sur l'Apprentissage automatique (CAP 2019), Jul 2019, Toulouse, France.
- Elies Gherbi, Blaise Hanczar, Jean-Christophe Janodet, Witold Klaudel. An Encoding Adversarial Network for Anomaly Detection. 11th Asian Conference on Machine Learning (ACML 2019), Nov 2019, Nagoya, Japan. pp.1–16.
- Elies Gherbi, Blaise Hanczar, Jean-Christophe Janodet, Witold Klaudel. Deep Learning for In-Vehicle Intrusion Detection System. Neural Information Processing. ICONIP 2020. Communications in Computer and Information Science, vol 1332. Springer, Nov 2020, Bangkok, Thailand.

Part I

Context: Would you entrust your life to your car?

Chapter 2

The cybersecurity of autonomous vehicles

As introduced in the Chapter 1, the automotive industry has undergone a paradigm change towards increasingly connected and autonomous cars. Smart cars available today are vehicles equipped with systems providing connected and added-value features to enhance car users experience and improve car safety. Within the next few years, smart cars connectivity is expected to expand, and smart cars will become connected to other vehicles, pedestrians and their surrounding infrastructure through information exchanges via V2X communications ENISA (2019). Cybersecurity is a crucial aspect that will affect the evolution of smart cars. There have already been several research publications on attacks targeting intelligent vehicles. In Fig 2.1 More than 70% of the industrial control system (ICS) vulnerabilities disclosed in the first half of 2020 can be exploited remotely claroty (2020). The Transportation sector looks particularly exposed in the top five most vulnerable sectors. The expectations are that the numbers of transportations vulnerabilities will increases when the ITS are deployed at a large scale.

In this thesis, our main focus is to study the in-vehicle network intrusion detection system. The primary function we want to secure is the In-vehicle network communication and its functionalities, as it is mainly one of the most used ways to perform intrusions and attacks. This chapter review the literature of cybersecurity gradually applied to the IT domain to its application in the ITS, more specifically for the vehicle and future autonomous vehicles (the in-vehicle security part). We review the IDS literature and its taxonomy from the In-vehicle perspective.

2.1 In-vehicle Network communication and architectures

ITS is often associated with intelligent cars, whether they possess driver-assistance technology, or semi-autonomous, or even fully autonomous. Intelligent cars are a major component of ITS due to the sheer volume of personal vehicles on the roadway. Several elements bind the scope of this research. The first element is limiting this research to cars. The term “vehicles” would be too broad because it includes cars, drones, trains and a wide variety of other devices. In the following, we use the term car and vehicle interchangeably to designate an automotive car. We

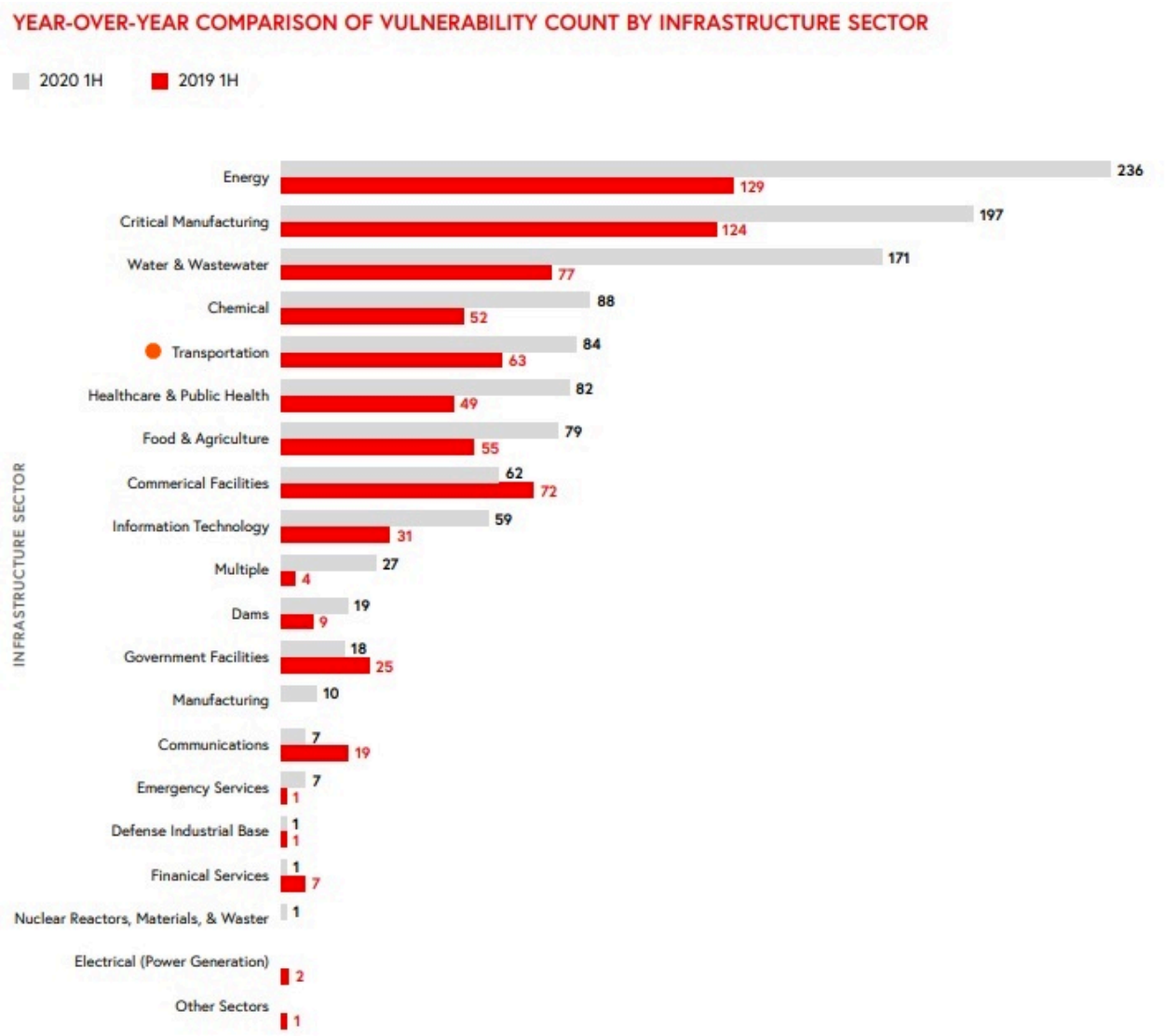


Figure 2.1: Industrial Control System (ICS) vulnerabilities disclosed in the first half of 2020. Credit claroty (2020)

will discuss and consider the assumptions and definitions related to all forms of connected cars. In chapter 1 we gave an introduction on ITS cybersecurity and in-vehicle network architecture. We extend the understanding of this introduction while defining this thesis's scope according to a taxonomy gathered from the literature review on connected vehicles, in-vehicle architectures, investigate the different attack vector and cybersecurity defence applied to the vehicle in general and the transferability of security from IT to automotive domain in the below sections.

We are interested in network security for in-vehicle systems. The vehicular network communication role is to support the required cooperation of different components of the vehicle.

We refer to all communications between the internal components of the vehicle as internal vehicular communication. Other communication technologies exist that provide a communication interface for external devices to perform diagnostic or firmware updates and future applications such as ITS, enabling connected cars to perform

traffic optimisation activities by exchanging messages between different autonomous vehicles. Moreover, functionalities that give passengers the ability to stay connected to the internet are getting more popular. We refer to this kind of communications and all communications that include an outside party as external communication.

2.1.1 External communication

External communication technologies are used to control several features of the car remotely, and connectivity in modern vehicles has become a necessity. Manufacturers are trying to give the consumer more ways to remotely control several aspects of the vehicle Koscher u. a. (2010a) using more than the traditional radio-controlled door unlocking functionality. For example, WiFi (IEEE 802.11) and Cellular communication such as GSM, 3G, 4G even the recent 5G are becoming a more standard option. These communication technologies are used for turning on air conditioning and even starting the engine. GPS for navigation and Bluetooth for hands-free smartphones usage has been used in the past decade Lee und Gerla (2010). Additionally, from the manufacturers' point-of-view, diagnostics messages are required to push a remote Firmware update to ensure that the customer gets the best aftermarket experience. The manufacturer may also request to receive a periodical report about the vehicles to provide remote support. These communications are usually done through VPNs provided by the manufacture or third-party services Shavit u. a. (2007).

More sophisticated services and communications features are incorporated into vehicles. On-Board Diagnostics (OBD-II) port provides direct, standard access to internal automotive networks for user-upgradable subsystems such as audio players who are routinely attached to these same internal networks. Telematics systems provide value-added features such as automatic crash response, remote diagnostics, and stolen vehicle recovery over a long-range wireless link. To do so, these telematics systems integrate internal automotive subsystems with a remote command centre via a wide area cellular connectionKoscher u. a. (2010a). Increased external communication between Autonomous Vehicles and the external environment: One main type of communications is the inter-vehicular, Vehicle-to-Vehicle (V2V) communications on the road Vehicular Adhoc Networks (VANETs). It allows information-sharing among nearby autonomous vehicles so that each car is better aware of its rapidly-changing surroundings Kumar u. a. (2012); Thing und Wu (2016). In future, V2I and vehicle-to-Internet of Things (V2IoT) communications will also become more prevalent on the roads. V2I communication is a wireless and bidirectional exchange of information between vehicles and road infrastructures, such as overhead RFID (Radio-Frequency Identification) readers and cameras, lane markers, traffic lights, street lights, road signs, and parking meters. As smart vehicles travel along roadways, On-Board Units (OBUs) within intelligent vehicles communicate with Road-Side Units (RSUs), IOT devicesHahn u. a. (2019); Poudel und Munir (2021); Lu u. a. (2014); Sun u. a. (2016).

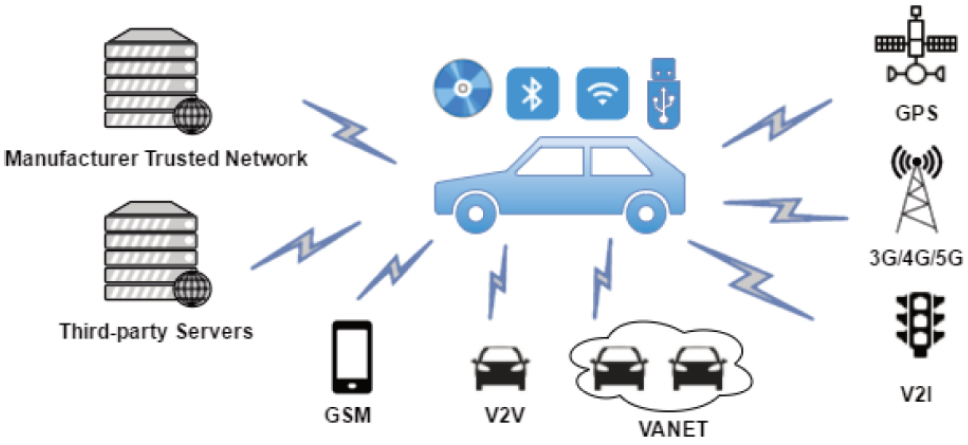


Figure 2.2: External vehicle communication

2.1.2 Internal communication

The in-vehicle network involves communications among different Electronic Control Units (ECUs), sensors and actuators. Almost all functions in the modern car are controlled by one or more. The number of ECUs in an automobile ranges from 30 for simple cars to approximately more than 100 for modern vehicles. The expectation of the gradual introduction of Autonomous vehicle from no automation, to conditional automation to full automation will take over more functions. When automation levels grow, the need for more processing power and sensors increases, also associated network bandwidth. It certainly may also lead to new car networks architecture and associated future ECUs.

Electronic Control Units (ECUs)

Nowadays, ECUs are cost-effective solutions. This technology implies a voltage rating of the microcontroller of 5 V or 3.3 V for the interface signalling and evens lower for the core supply. The motivation for these types of products is system cost reduction. It is technically feasible since the integrated function requirement uses the same high voltage technology, such as engine control, window lifter, or electronic power steering. There is some combination like braking and airbag ECU. Moderns and future in-vehicle networks call for multiple domains functionalities. In this case, the communication must support applications with data ranging from $125Kb/s$ to $1Mb/s$, application demanding high bandwidth and strict latency constraint like high-resolution video entertainment system and Advanced Driving Assistance System (ADAS) ($10Mb/s$ to $30Mb/s$).

In-vehicle networks were realized through point-to-point wiring between electronic components, resulting in bulky, expensive, complicated harnesses Nilsson u. a. (2008); G. Leen u. a. (1999). With increasing scale and complexity of the in-vehicle network and the grow where volume, weight, and reliability become real problems. As a result, several automobile communication bus protocols are developed (See Section 1.3.1). By connecting a number of the

electronic components to the same in-vehicle communication bus, sharing the communications medium and wiring can thus be saved. Meanwhile, the in-vehicle architecture can be more hierarchical and structural, simplifying the automobile design procedure. Many standards are used in the automobile industry, particularly the networks and their protocols based on their local requirement and needs while keeping it cost-effective. Due to different cost, reliability, bandwidth, latency requirements imposed by different automobile applications classifications, specific communication protocols, and networks have been developed (LIN, CAN, FlexRay, Ethernet) and many network architectures have been proposed Dibaei u. a. (2019); nxp.

In-vehicle network architectures

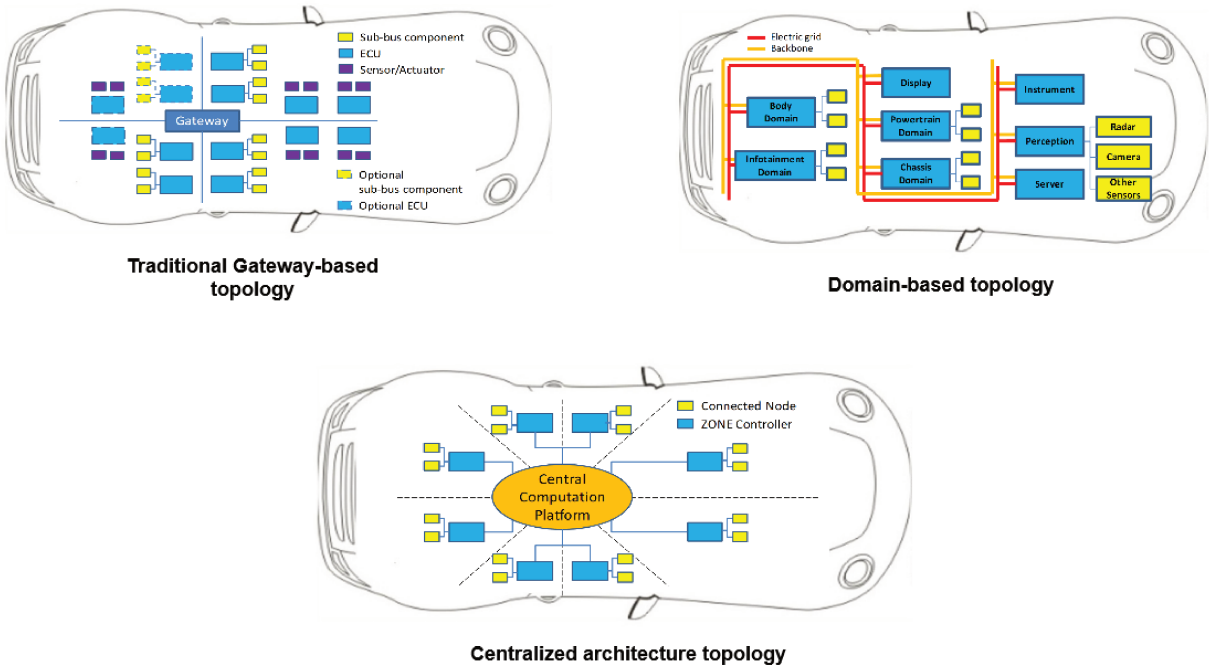


Figure 2.3: Intelligent Vehicle System Architecture Dibaei u. a. (2019)

Integrating all the new modern cars functionalities is not feasible by putting them together using traditional automotive architecture design. Indeed, those functionalities can be classed into different application types based on safety, priority, and other resources required. In Fig 2.3 we show three topology design of in-vehicle network architecture, 1) The traditional topology is based on the controller area network (CAN). Due to the characteristics of CAN, every node in the network must share the bandwidth. The bandwidth is like a bottleneck that limits the data processing ability of each ECU on the network. The traditional architectures core problem is the lack of space for the high computation power unit, which is necessary for intelligent driving. 2) Domain-based topology concept is to divide the autonomous driving system into several ECUs domains based on each domain's core computation. The vehicle components can be classified into different domains according to their functionalities. Usually, the sensors

and actuators different functionalities can share those would be grouped as one domain. This topology can support more complex intelligent driving functions, as each domain ECU has more power in both communication and computationHaas und Langjahr (2016). 3) In a centralized architecture, most of the computation tasks are realized in the central computation entity. Most of the components are connected to the central computation entity to access all sensors and actuators data. This topology enables the combination of more information, thus enhance the potential of making a better decision. However, a centralized topology has higher demands on the data communication capacity, so it needs to group the components into different sub-networks according to their physical placement or the network properties to improve the efficiency of communication Brunner u. a. (2017).

2.1.3 Controller Area Network (CAN)

As mentioned above, one of the most critical challenges for the next generation in-vehicle architectures is managing the high-speed communication among vehicles electronic components with a limited cost. The most successful communication network in the current automotive industry is the Controller Area Network (CAN) protocol. CAN is without any doubt the most widely used standard in the field of vehicle hardware communication since its publication in 1986. CAN is a network protocol developed by Robert Bosch for vehicle systems Szydlowski (1992). The original specification paper from Szydlowski (1992) points out many proprieties that led to the widespread adoption of this protocol standard (Prioritization of messages, Guarantee of latency times, Multicast reception with time synchronization, System-wide data consistency, Multimaster, Error detection and signalling, Automatic re-transmission of corrupted messages). Compared to other network technologies, CAN have two outstanding advantages: cost efficiency and flexibility Zeng u. a. (2016); Afsin u. a. (2017); Hartwich und Bosch (2012). Effectively, CAN is a priority-based bus implemented by using two wires. The Medium Access Control (MAC) protocol of CAN uses carrier sense multiple access with collision detection (CSMA/CD). Up to 8 bytes of data can be carried by one CAN frame, and a cyclic redundancy check (CRC) of 16 bits is used for transmission error detection. CAN facilitates bit by bit non-destructive arbitration over the identifier which also serves as a priority flag.

In the following, we will present a brief explanation of CAN, which was designed to be extremely reliable and flexible because it's meant to work in harsh environments such as vehicles. CAN operates in two layers of the OSI model. **The physical layer** which is used for signalling, and **data link layer** that handle several aspects of the transmission of CAN frames, the message filtering and recovery management.

The physical layer

The physical layer of CAN is concerned with defining how to send signals and the correct encoding of bits. Thus, several concepts are specified in the original CAN specification :

(A) **Dominant and recessive logical bus states :** CAN use two distinct wires to transmit signals: the CAN High (CANH) and CAN Low (CANL) wire. When no signals are transmitted on the bus, these wires are set into an idle state; the voltage amounts to $2.5V$. After the first bit has been sent, the CANH wire increases its voltage to $3.75V$, and the CANL wire decreases its voltage to $1.25V$. So, the transmission of data in the CAN bus is through differential signalling and what this means is that we use both wires (CAN high and CAN low) to transmit data at the same time. Because the objective is the difference in voltage between these two wires as you can see from the Fig 2.4, the recessive level means that both of the wires will stay at around $2.5V$, so the difference between the two wires is close to zero. The dominant level means that the CAN high wire goes to a higher voltage, whereas the CAN low wire goes to a lower voltage, creating a $2.5V$ difference between the two wires. We note that the recessive level is the logical one, and the dominant level is the logical zero Natale u. a. (2012).

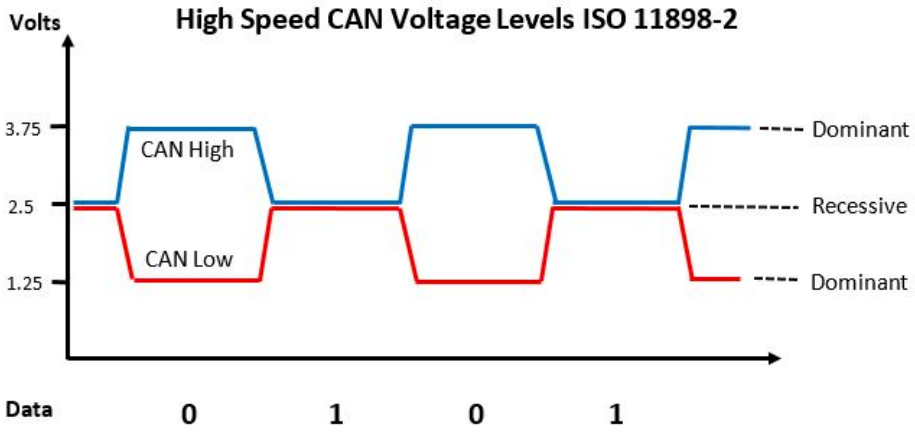


Figure 2.4: Dominant and recessive logical bus states pico technology

(B) **Bit representation :** CAN uses Non-Return-to-Zero (NRZ) bit encoding. This encoding describes that after a value of 1 is detected in the bitstream, the next bit does not have to be changed to a 0 immediately and the voltage can be maintained for a longer period (See Fig 2.5). This encoding can lead to a desynchronization of the communication. All participating nodes are synchronized and adjusted to the same clock rate to ensure the accurate transfer of data. A sovereign clock signal does not achieve the clocks synchronization, but the CAN frames on the bus are utilized. All nodes connected to the CAN bus listen for frames and synchronize their internal clocks to the transmitting node clock Natale u. a. (2012).

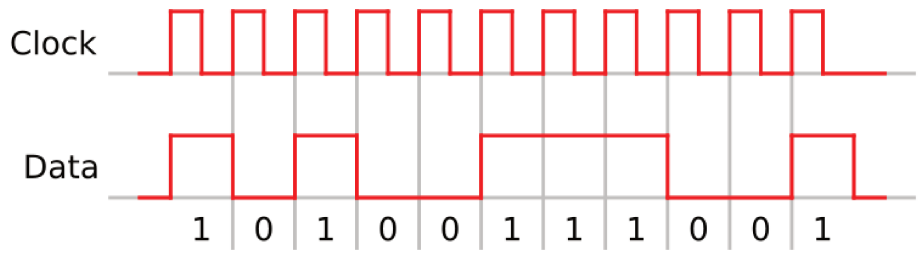


Figure 2.5: Non-Return-to-Zero encoding example with clock synchronized on the transmitting

(C) **Bit stuffing :** To prevent the communication desynchronization as mentioned above, the CAN uses a practice called Bit stuffing. The central idea behind bit stuffing is to inject a bit of reversed polarity, after five bits of equal polarity have been determined in the communication, thus enforcing the correct synchronization (See Fig 2.6). It important to mention that not all fields become stuffed, for instance, the CRC and the ACK (See Fig 2.7) have a fixed size and can not be stuffed Natale u. a. (2012).

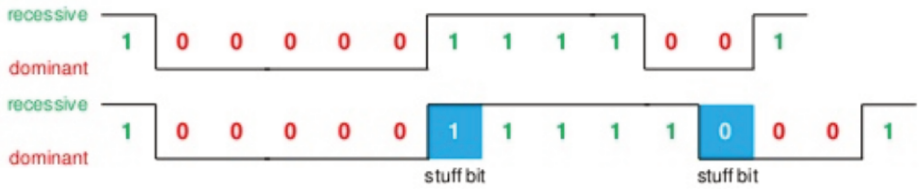


Figure 2.6: Bit stuffing example mbedlabs (2016)

Data link layer

CAN data link layer protocols have some features; we highlight some of them; any node has the right to request transmission rights at any time. The necessary bus arbitration method to avoid transmission conflicts. Broadcast transmission of the CAN data frame.

(A) **CAN frames :** The majority of the communication proceeds through data frames that constitute the data field, the arbitration field, Cyclic Redundancy Check (CRC) field, and the acknowledge field. The arbitration field further contains an 11-bit identifier field and a Remote Transmission Request (RTR) field used in the arbitration and must be set to a dominant bit in case of a data frameSzydlowski (1992). The data field then follows, which can be 8 bytes in total, followed by the cyclic redundancy check field. The structure of the data frame is illustrated in Fig 2.7.

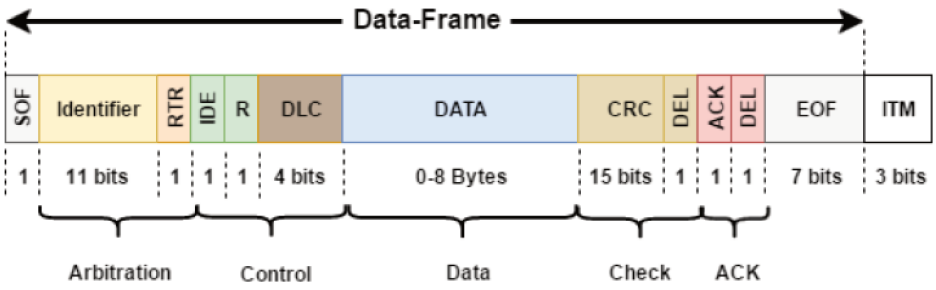


Figure 2.7: CAN frame Navet und Simonot-Lion (2013)

(B) **Message Arbitration :** Each message has been assigned an identifier frame to define the message priority. The lower number of message identification value, the higher priority it has to gain the bus (See Fig 2.8). This prioritization feature has also solved the bus access conflict so that if two nodes want to send data simultaneously, each ECU with a lower ID value will first publish it Avatefipour und Malik (2017); Corrigan (2002); Navet und Simonot-Lion (2013).

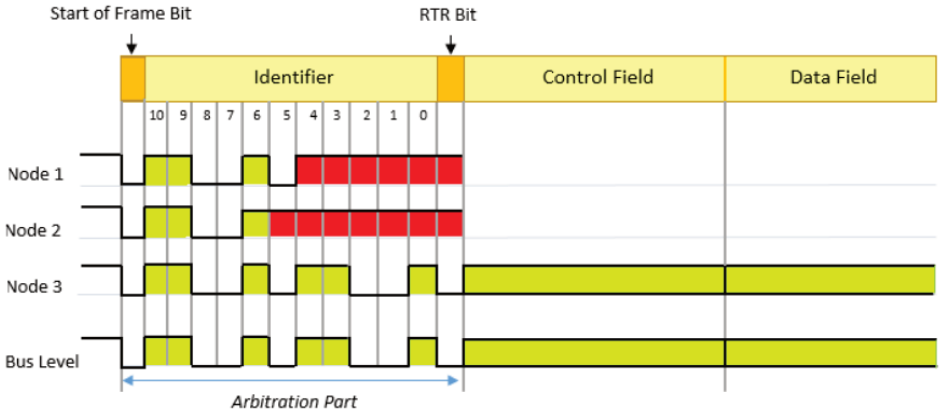


Figure 2.8: Arbitration process in CAN-Bus protocol.Avatefipour und Malik (2017)

2.2 Network Cybersecurity: A brief taxonomy

2.2.1 Intrusions

In order to provide useful services and allow a system (backed by people usage or autonomous) to perform tasks more conveniently, computer systems are linked to networks; the result is a worldwide collection of local and wide-area networks known as the Internet. While ease of use and convenience are tradeoffs with security, we often look at it as a risk challenge to overcome or mostly to minimize instead of renouncing to the services provided by remote machines and applications. Therefore, we have to deal with a loss of security.

When a computer system is attached to a network, one can identify three areas of increased risk: First, the

number of points that can potentially serve as the source of an attack against a computer increases Cheswick u. a. (2003). For a stand-alone system, physical access to the machine is a prerequisite to an intrusion. When a system is connected to a network, each host can be utilized by an intruder to send packets to the system. Certain services (such as web servers or name servers) need to be publicly available, each machine on the Internet might be the originator of malicious activity Kruegel (2004).

Second, the physical perimeter of the system is extended. For a single machine, everything is considered to be inside a box. The data is protected from tampering while transferred between the different components of the computer hardware. The same assumption is not valid for data transmitted over the network. Packets on the wire often pass areas and are forwarded by infrastructure devices that are entirely out of the control of the receiver Kruegel (2004).

Third, networked machines typically offer a more significant number of services than the single authentication service provided by a stand-alone system. All service processes implementing remote access may contain exploitable program bugs or configuration errors that can lead to a security compromise Kruegel (2004).

There are many types of intrusion, making it difficult to give a single definition of the term. Asaka u. a. (1999); Kruegel (2004); Engen (2010) offers the following processes of a successful intrusion:

Surveillance stage : The intruder attempts to learn and gather as much information as possible about the potential target by scanning communication, software's and tries to discover vulnerable services and configurations errors that can be exploited.

Exploitation stage: Once the attacker identifies weaknesses in the previous stage, they can elevate the privileges. Then, the intruder will have free access to abuse the system targetEngen (2010).

Mark stage: After the exploitation stage, the attacker may be free to implement the action(s) wanted, steal information from the system, destroy data, plant a virus or spyware software, change values or use the host as a medium for conducting further attacks. After which, this marks the stage where the attacker has achieved his goal(s)Asaka u. a. (1999).

Masquerading stage: Finally, the masquerading stage covers all activity performed by an intruder after the successful break-in (e.g., deleting log entries or patching the vulnerability used to get access).

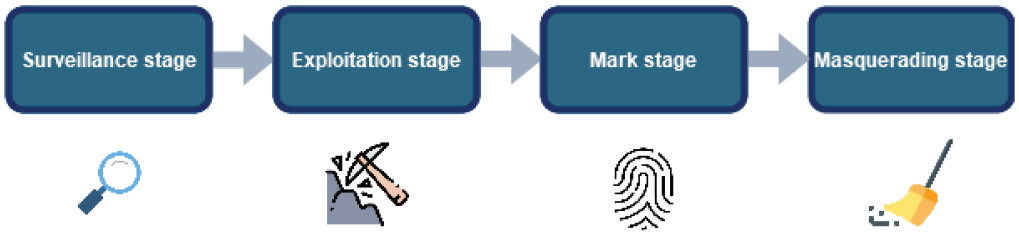


Figure 2.9: Schema showing the different steps of an intrusion.

2.2.2 Network Threats, Attacks and Security Properties

In the purpose of network security, we assume that a system function is to provide information. In general, there is a flow of data from a source (e.g., a host, a file, memory, subsystem, node in distributed systems) to a destination (e.g., a remote host, another file, a user or a subsystem) over a communication channel (e.g., a wire, a data bus, WIFI) mainly governed by a communication protocol. The security system's task is to restrict access to this information to only those parties that are authorized to have access to achieve a specific functionality, according to the security policy in use Kruegel (2004). A regular course of communication is a legitimate flow of data between the concerned source and destination while respecting security proprieties.

Threats

In general terms, an intrusion threat is an unauthorized attempt to access valuable assets, manipulate information, and alter a systems state into instability or unreliability. Those it causes a deviation from the expected use of the system Koscher u. a. (2010b).

We show the regular communication flow and several types of threats models that target it in Figure 2.10 and their description below.

- **Interruption:** An asset of the system gets destroyed or becomes unavailable. This attack targets the source or communication channel. It prevents the information from reaching its intended target. Attacks in this category attempt to perform a kind of Denial Of Service (DoS)Peng u. a. (2007); Kruegel (2004); Loukas und Öke (2010); Mahjabin u. a. (2017).

- **Interception:** An unauthorized party gets access to the information by eavesdropping into the communication channel to capture data being transmitted on a network (performing sniffing or snooping attack) while it can keep traffic flowing efficiently Anu und Vimala (2017); Philip Baczwewski (2000); Prowell u. a. (2010).

- **Modification:** The information is not only intercepted but modified by an unauthorized party while in transit from the source to the destination. Man-In-The-Middle (MITM) attacks occur when unauthorized individuals or parties are placing themselves in the path of communication to eavesdrop, intercept, and possibly modify legitimate communications. Prowell u. a. (2010); Kruegel (2004)

- **Fabrication:** An attacker inserts counterfeit objects into the system without having the sender doing anything. When a previously intercepted object is inserted, this processes is called replaying Patil und Kamble (2018). When the attacker pretends to be the legitimate source and inserts her desired information, we call it masquerades. Counterfeiting attack takes place when an attacker forges the message that can be scanned and received by authorized readersPhilip Baczwewski (2000); Rong u. a. (2013); Patil und Kamble (2018); Kruegel (2004).

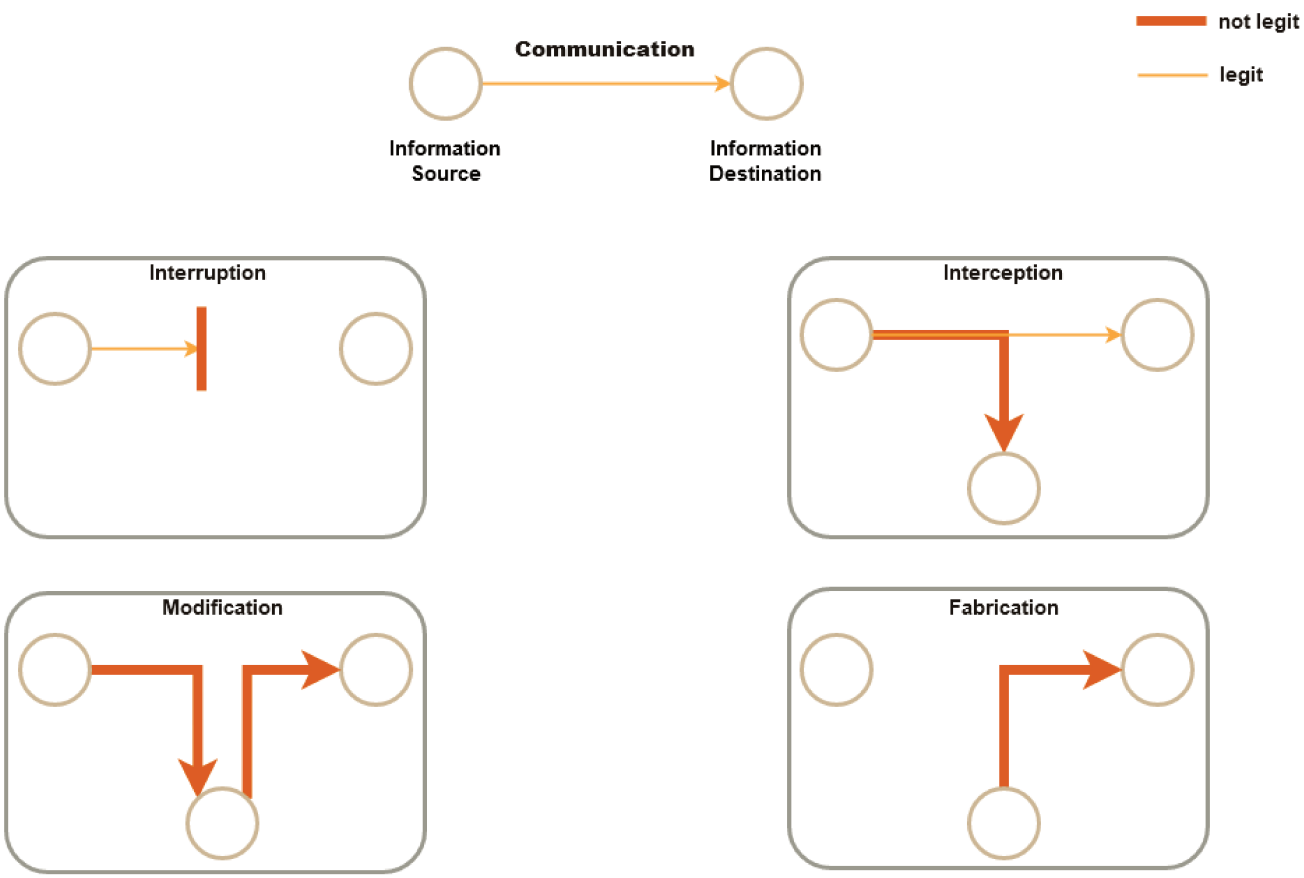


Figure 2.10: Schema of different threats (unauthorized Network manipulations) Inspiration credit Kruegel (2004)

Network attacks

It is essential to describe a threat model to implement a layered and holistic security mechanism. Once a threat model is defined, attacker types and attack vectors can be derived according to the capabilities of adversaries, their possible intentions and the valuable assets in the target network. The threat models are valuable abstraction of possible attacks, yet, it is nearly impossible to cover the complete space of attack types in real life.

We present various types of network attack classified into following three categories given by Bijone (2016a).

- **Denial Of Service (DoS) :** A Denial of Service attack attempts to slow down or completely shut down a target to disrupt the service and deny the legitimate and authorized access for users or applications. Peng u. a. (2007); Loukas und Öke (2010); Mahjabin u. a. (2017). Such attacks are widespread on the Internet where a collection of hosts are often used to bombard web servers with dummy requests. There are several different kinds of DoS attacks Bijone (2016a) we present some of them. *Flooding DoS Attacks* an attacker sends more requests to a target that it can handle. Such attacks can either exhaust the target's processing capability or exhaust the network bandwidth of the target, either way leading to a denial of service to other users Peng u. a. (2007).

  Distributed Denial of Service attack (DDoS), is using a large pool of hosts to target a given victim host. Once an important number of hosts are compromised, the intruder instructs them to launch various flooding attacks against a specified target Mirkovic und Reiher (2004); Kumar und Vajpayee (2016).

- **Penetration Attacks :** In penetration attack, an attacker gains an unauthorized control of a system and can modify/alter system state. Generally, such attacks exploit certain software flaws, enabling the attacker to install viruses and malware in the system. The most common types of penetration attacks are:

  User to Root (U2R), U2R is an attack that aims to gain superuser access to the system. Attacker gains superuser access by exploiting vulnerabilities in the operating system or application software. The attacker starts with access to a normal user account on the system and can exploit some vulnerability to gain root access to the system Bijone (2016a); Kendall (1999). Improving the detection rate of U2R attack classes is an open research problem. The most common attack in this class of attack is Buffer overflows. Buffer overflows occur when a program copies too much data into a static buffer without checking to ensure that the data will fit Kendall (1999).

  Remote to Local (R2L), R2L is an attack in which the attacker tries to gain unauthorized access from a remote machine into the local target system. Hence, gains access to the inaccessible files stored locally on the host. There are some similarities between this class of intrusion and U2R, as similar attacks may be carried out. However, in this case, the intruder does not account for the host and attempts to obtain local access across a network connection. To achieve this, the intruder can execute buffer overflow attacks, exploit misconfigurations

in security policies or engage in social engineering Kendall (1999); Bijone (2016a)

- **Scanning Attack :** In such attacks, an attacker sends various kinds of packets to probe a system or network for a vulnerability that can be exploited. When probe packets are sent, the target system responds; the responses' analysis determines the target system's characteristics and vulnerabilities. Thus scanning attack essentially identifies a potential victim.Bijone (2016a); Kendall (1999); Engen (2010). An attacker with a map of which machines and services are available on a network (network topology, type of firewall, identifying hosts that respond, operating systems and server applications running) can use this information to look for weak points. Some of these scanning tools enable even a very unskilled attacker to very quickly check hundreds or thousands of machines on a network for known vulnerabilitiesKendall (1999). Scanning is typically considered a legal activity and there are a number of examples and applications that employ scanning. The most well-known scanning applications are Web search engines.

Security Properties

Before one can evaluate attacks against a system and decide on appropriate mechanisms to defend these threats, it is necessary to specify a security policy Tanenbaum und van Steen (2002). A security policy defines the desired properties for each part of a secure computer system. It is a decision that has to consider the value of the assets that should be protected, the expected threats, and the cost of proper protection mechanisms Kruegel (2004); Tanenbaum und van Steen (2002). A sufficient security policy for a regular computer user's data may not be sufficient for a bank. The security policy that is sufficient for a bank may also not be sufficient for an ITS or an in-vehicle network system. Indeed, the context of future autonomous transportation could be the more likely target and has to protect vital resources and passenger's safety (Section 2.1).

The threat mentioned above violates the different security policies of the computer system. A security property describes a convenient feature of a system concerning a certain type of threat. A common classification is given by Kruegel (2004); Coulouris u. a. (2011); Northcutt (1999); Tanenbaum und van Steen (2002) listed below:

- **Confidentiality:** This property covers the protection of transmitted data against its release to unauthorized parties. In addition to protecting the content itself, information communication should also be resistant against traffic scanning and whereby its information is disclosed only to authorized parties.

- **Integrity:** Is the policy that protects the information transfer against modifications. This property guarantees that a single message arrives the receiver as it has transmitted by the sender, but integrity also extends to a stream of messages. It means that no messages are lost, duplicated, or reordered, and it makes sure that the messages cannot be replayed. As destruction is also covered under this property, all data must arrive at the receiver. Integrity is essential as a security property and as a property for network protocols. It must also

ensure the message integrity in case of random faults, not only in malicious modifications, which means that any improper alterations in the communication flow should be detectable and recoverable.

- **Availability:** It defines a system whose resources are always available for usage in the limit of their capacity. Whenever information needs to be transmitted, the communication channel is available, and the receiver can cope with the incoming data. This property makes sure that threats cannot prevent resources from being used for their intended purpose. A highly available communication system will the most likely working at a given instant in time.

- **Authentication:** Is about making sure that the information is authentic, which means that it verifies the sender's claimed identity. Authentication property assures the receiver that the message is from the source that it claims to be. So, it makes sure that no third party cannot pretend successfully being another source.

- **Non-repudiation:** This property describes how to prevents either sender or receiver from denying a transmitted message. When a message has been transferred, the sender can prove that it has been received. Similarly, the receiver can prove that the message has actually been sent Kruegel (2004).

2.2.3 Security Defences and mechanisms

To defend against the threat mentioned above, and respect the security policy, different security mechanisms can enforce the security properties defined. Depending on the anticipated attacks, different means have to be applied to satisfy the desired properties. Kruegel (2004); Tanenbaum und van Steen (2002) propose three main classes of measures against attacks: attack prevention, attack avoidance, and attack detection.

Attack Prevention

Attack prevention is a way of preventing certain attacks before reaching and affecting the target. An essential factor in this class is access control, a mechanism which can be applied at different levels such as the operating system, the network, or the application layerTanenbaum und van Steen (2002). Access control limits and regulates access to critical resources. A firewall Cheswick u. a. (2003) is a vital access control system at the network layer. It prevents attacks from the outside against the inside network machines by denying connection attempts from unauthorized parties located outside.

Attack Avoidance

Based on the possibility that an intruder may access the target resources, to protect the confidentiality and the integrity of the information, attack avoidance mechanism modifies the information to make it unusable for the attacker. The data is preprocessed at the sender before transmitted over the communication channel and post-processed

at the receiver. It resists attacks by being nearly useless for an intruder. If no modification occurs, the receiver's information is identical to the sender's one before the preprocessing step. The most important tool used in this class is cryptography Schneier (1995). It allows the sender to transform information into what may seem like a random data stream to an attacker but can be easily decoded by an authorized receiver.

Attack Detection

Attack detection assumes that an attacker can obtain access to the desired targets and successfully violate a given security policy. When undesired actions occur, attack detection has the task of reporting that something went wrong appropriately. Also, it is often desirable to identify the exact type of attack. An essential aspect of attack detection is recovery. Often it is enough to report that malicious activity has been detected. Still, some systems require that an attack be reverted or stopped. The attack detection operates under the worst-case assumption that the attacker gains access to the communication channel and can use or modify the resource. The most used tool of the attack detection class are intrusion detection systems Kruegel (2004). Because this thesis focus on intrusion detection systems, the remaining sections of this chapter are dedicated to a more detailed introduction to intrusion detection, and its applications in the ITS domain.

2.3 Intelligent In-vehicle network Cybersecurity

2.3.1 Objective

We present two separate vehicle security concerns (In-vehicle security and VANETs security), In this work, the external communication is considered as an attacking interface.The VANETs securiy network Bariah u. a. (2015); Engoulou u. a. (2014); Hasrouny u. a. (2017) is getting a lot of attention in the research era. Nonetheless, this thesis scope is in-vehicle network intrusion detection (See Fig 2.11).

The same as security in internet and communication technologies IT, In-vehicle network security objective is facing the threats by meeting the underlying standard security requirements introduced above (See in Section 2.2.2). The preservation of these objectives intends to secure the in the in-vehicle network communication. These security principles are defined in the following with the perspective of vehicle networks El-Rewini u. a. (2020); Hahn u. a. (2019); Dibaei u. a. (2019); Chattopadhyay u. a. (2020); Qiang Hu (2018).

- **Confidentiality :** Confidentiality requires the content of a message not to be disclosed to any parties other than the intended ones. Lightweight Encryption mechanisms are proposed for in-vehicle networks to satisfy this objective Mundhenk u. a. (2017); Radu und Garcia (2016). Recent works in steganography and covert channels have investigated how those methods can conceal information when malicious actors have access to the communication channel Manchanda und Singh (2015).

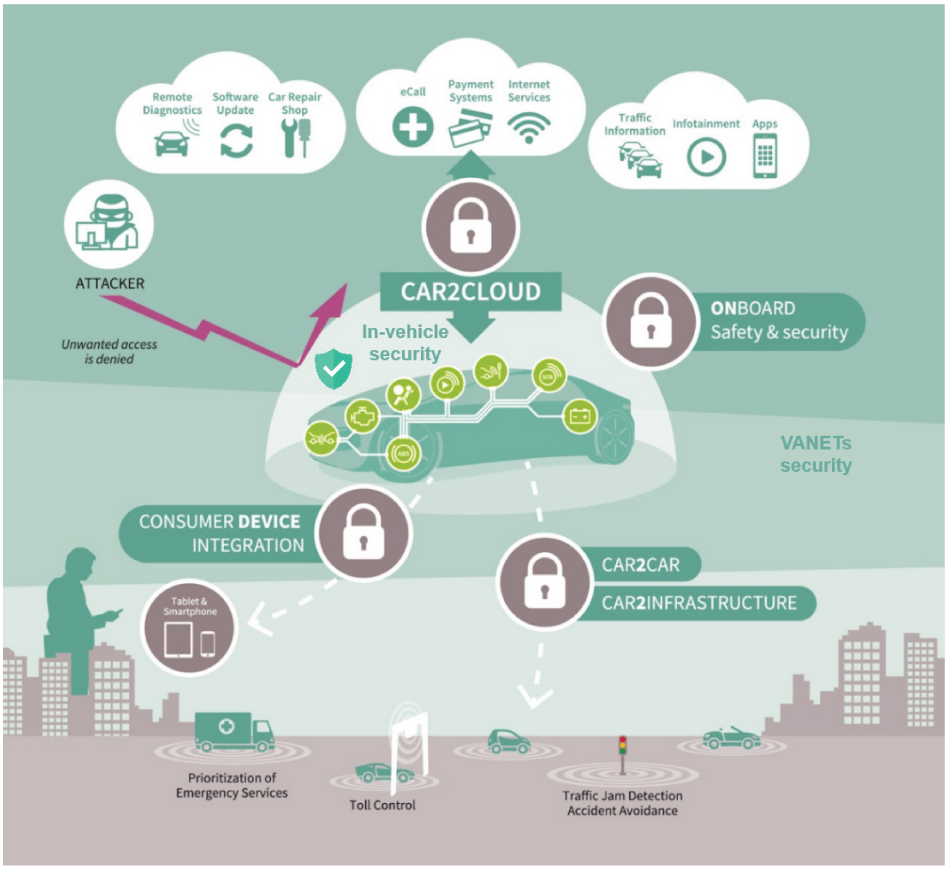


Figure 2.11: In-vehicle network security boundaries Vs VANETs security in the context of intelligent transportation systems.credit CYPRESS (2018)

- **Authenticity :**

  One of the primary objectives in-vehicle communication is to ensure only trustworthy entities in the networks. Authenticity is being able to verify the sender of any message in the system, so the data can only be accessed by authorized ECU, in the case where an attacker has access to the network with an external device Koscher u. a. (2010a); Forest und Jochim (2011); Groll und Ruland (2009).

- **Integrity :**

  To maintain the accuracy and completeness of message content is the objective of Integrity principle. Public Key Infrastructure (PKI) solutions are proposed for vehicle communication to verify whether the received data is corrupted or legitimateOthmane u. a. (2015).

- **Availability :**

  Availability ensures timely and reliable access to and use of information for nodes in the in-vehicle network. To preserve the functional state of ECUs in a vehicle network and complement the security requirements.

- **Non-Repudiation :** Any broadcasting node should not deny the right to authenticate. This property is fundamental in the case of an accident. After the accident, the driver must be rightly identified during the investigation, and before the accident, every message should be transmitted reliably El-Rewini u. a. (2020).

- **Real-Time constraints :** Outdated information is of no use in the high mobility environment of a VANET or in-vehicle communication. Ancient weather or traffic information is not useful, especially for autonomous driving systems; therefore, it must be prevented from delayed transmission El-Rewini u. a. (2020).

In-vehicle communication occurs within automotive bus systems, enabling message transmission between vehicle ECUs. Vulnerabilities exploitation of an in-vehicle network can lead to severe issues such as critical ECU reprogramming and taking control of the vehicle over the Controller Area Network. Those attempts cause the violation of the in-vehicle security principles listed above.

We consider the different step of an intrusion introduced in Section 2.2.1. The kill chain consists of four stages Fig (See 2.9). The attacker must pass all four steps to achieve a successful attack on the connected vehicle; to this end, the attacker uses three necessary elements. The first element is finding Point access. The second element is compromising an ECU, and the third element is finding a control feature that could be compromised. The number of technical vulnerabilities indicates the feasibility of different kind of attacks. In Checkoway u. a. (2011); Avatefipour und Malik (2017); Thing und Wu (2016); Dibaei u. a. (2019); Petit und Shladover (2014); Petit u. a. (2015); Toyama u. a. (2018) the authors present a broader view how an attacker can compromise vehicle security.

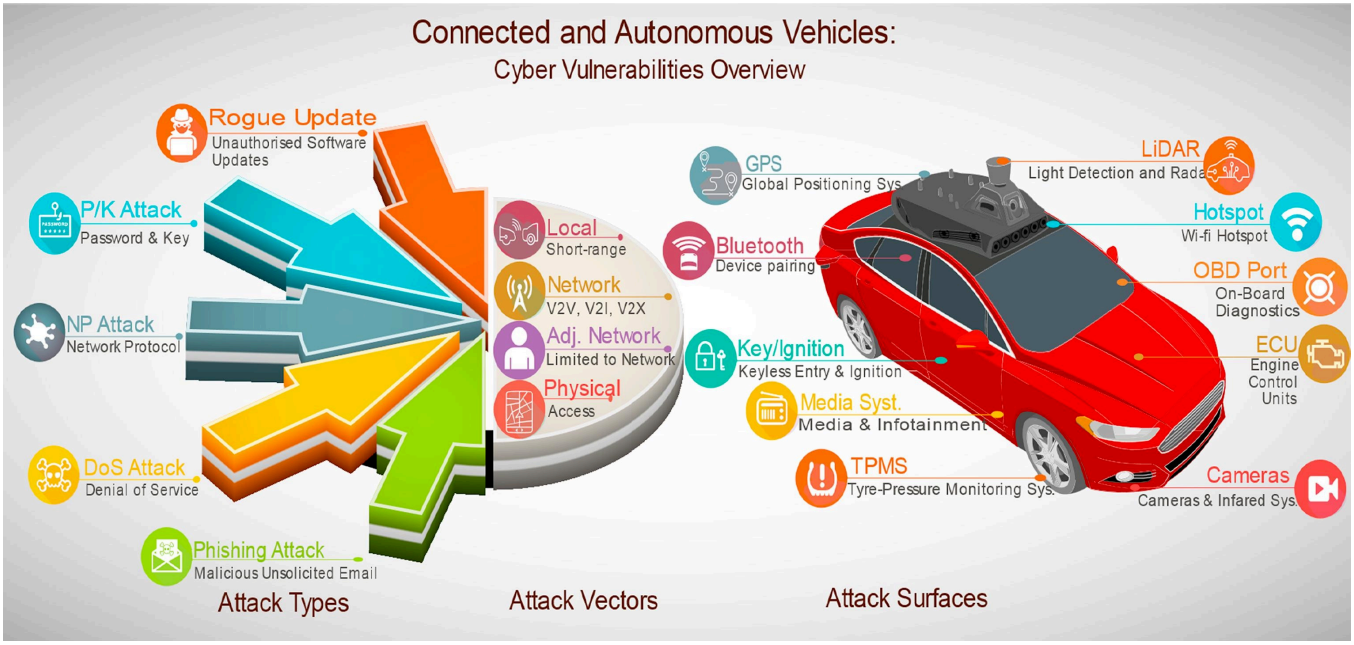


Figure 2.12: Overview of Threat and attack interfaces of an in-vehicle network system.credit Sheehan u. a. (2019)

2.3.2 Threat and attack interfaces of an in-vehicle network system

The first step of an automotive attack is to exploit an external interface to access the vehicle's internal systems. In the Section 2.1.1 several external interfaces have been described, in this part, we describe how an attacker can leverage these interfaces to gain access to the in-vehicle network. The interfaces present on modern cars have different ranges, from physical access to remote access. The Fig 2.12 shows the identified attack interfaces. The first class is the direct physical attack possibility. In that case, the attacker has direct access to all parts of the vehicle. The second class is about the remote attacks on a connected car. Interfaces with a longer range generally have a convenient aspect for an attacker as it is easier for the attacker to preserve the connection during the full attack.

Physical Access

Inside a modern and future automotive vehicle, there are multiple physical interfaces, and some are directly connected to the internal network.

OBD-II is the most well-known connector and is used by many security researchers to find and execute automotive attacks. In addition to monitoring electrical failures, the second-generation OBD also monitored emission-related systems and provided standardization across different manufacturers Carsten u. a. (2015). OBD-II ports are vulnerable to in-vehicle network access attacks and dongle exploitation attacks.Miller und Valasek (2014) were able to transmit and receive messages over CAN using an ECOM cable and homemade connectors to connect to the OBD-II port.

USB ports have become prevalent in modern-day vehicles, since they can connect phones, navigation systems, and USB devices to the vehicleOnishi u. a. (2017). Cai u. a. (2019) found that attackers could use the USB port to create a backdoor within the BMW Next Best Thing (NBT) vehicle entertainment system.

For electric vehicles, another physical entry point to the in-vehicle network is the charging infrastructure. Eventually, the charging infrastructure could be used to conduct attacks Bernardini u. a. (2017). Many attacks on electric vehicle charging have been identified. Mustafa u. a. (2013); El-Rewini u. a. (2020) charging is susceptible to masquerading, tampering, eavesdropping, and denial of service attacks, in addition to privacy concerns and charging thievery. In Fries und Falk (2012), the author discussed the man-in-the-middle and tampering attacks on the payment price and the amount of energy that the meter believes the electric vehicles has received. many other threats related to electric cars has been demonstrated and discussed in Sun u. a. (2015); Alcaraz u. a. (2017); Vaidya und Mouftah (2018); Lee u. a. (2014).

The infotainment system inside a car provides the other physical access points, through discs and USB drives. Often the infotainment system is connected to the CAN bus. The information supplied by infotainment systems can include (voice calls, text messages, emails, social networking, personal contacts) and other forms of data that can be received by connecting to a mobile phone. Infotainment system vulnerabilities were demonstrated when the BMW ConnectedDrive infotainment system was hacked because of its corresponding in-vehicle Network gatewayEl-Rewini u. a. (2020); Robinson-Mallett und Hansack (2015).

The attacks requiring physical access can be both invasive and non-invasive, *sidee-channel attacks* are non-invasive attacks that refer to revealing useful information regarding the transmitted data in the in-vehicle network. On the other hand, invasive attacks are allowed through physical access that enables the intrusion to the vehicle bus system and its ECUs, resulting in *code modification, code injection, packet-sniffing and fuzzing* Thing und Wu (2016).

Remote Access

Remote access is more convenient for the attacker. Some remote access interface (short-range) requires the attacker to be nearby the vehicle during the attack's entire duration. The range of these kinds of access points is typically about 10 metres to 300 metres for some access points(Bluetooth, Pressure Monitoring System(TPMS), Wi-Fi hotspot, V2X). Long-range interfaces (radio, GPS, Cellular) has a more extended range superior to 10km Francillon u. a. (2010); Rouf u. a. (2010). Although this requires enhancement misappropriation techniques.

Telematics systems complement infotainment systems by providing information on internal vehicular systems, which includes (fuel efficiency, engine failures, brake pad wear, transmission). Jo u. a. (2017) identifies security risks in Android OS-based telematics systems that enabled drivers to remotely unlock and lock car doors, start and stop the car engine using low-speed CAN, and access diagnostic information using high-speed CAN.

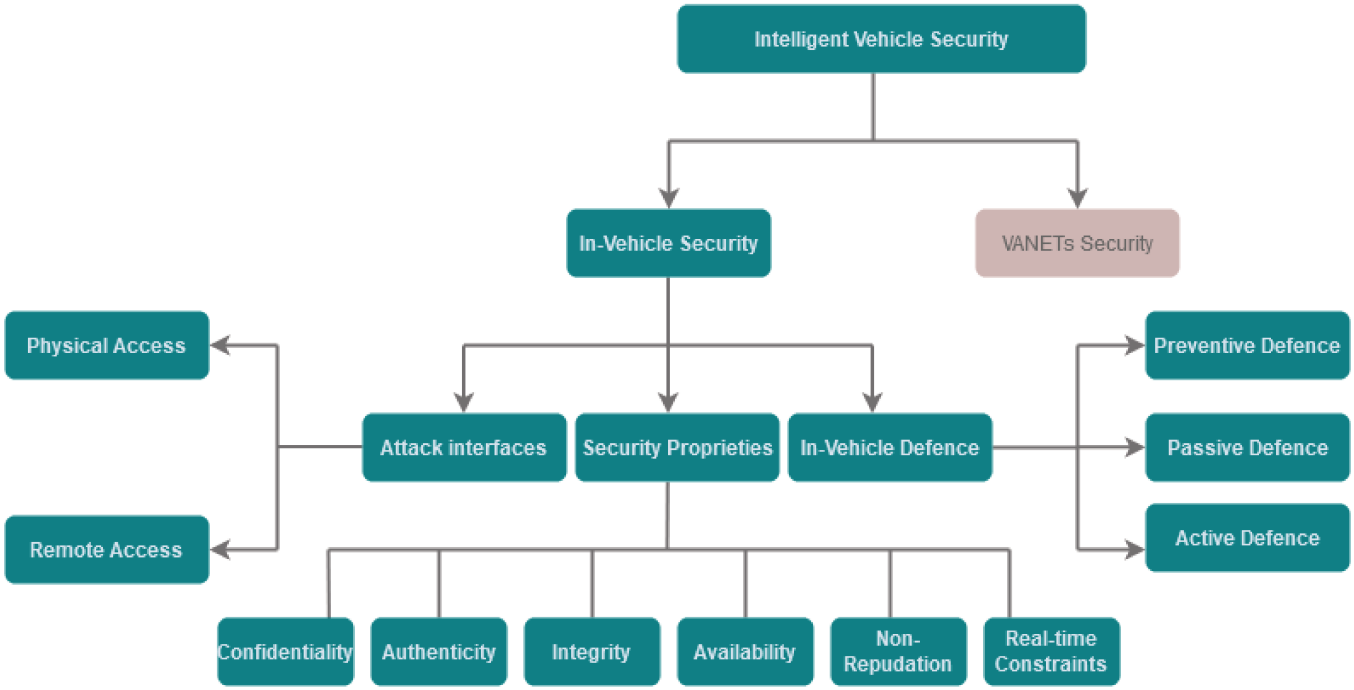


Figure 2.13: A brief taxonomy of in-vehicle security.

In short-range communication, Bluetooth is often used to pair smartphones to vehicles infotainment and telematics systems. in Onishi u. a. (2017) the author demonstrates a listening vulnerability on Bluetooth, that enables *buffer-overflow* attacks. in Seri und Vishnepolsky identified a new attack vector known as BlueBorne, which spreads wirelessly and attacks devices by exploiting their Bluetooth vulnerabilities.

Wi-Fi and WiMAX (IEEE 802.16 standard) is also a candidate for V2V and V2I communication, with low-latency, Quality of Service (QoS), security features and all-IP core network support citeinvecsec24. In Sen Nie (2017), the authors could remotely hack a Tesla Model S vehicle, by exploiting the fact that the password to an embedded Wi-Fi Service Set Identifier (SSID) was saved in plain-text. They were then able to fake a hotspot and redirect traffic to their domain.

Attackers may exploit the vehicles enhanced connectivity vulnerabilities, to gain access to the in-vehicle network system remotely. Especially the wireless connection and external-facing sensor interfaces such as its LiDAR, camera and GPS. But it also enables the attacker to perform attacks on the external communication VANETs.

2.3.3 In-vehicle Defence Mechanism

Once the attacker had succeeded in taking over the entry point via physical or remote access, the second element is to compromise an ECU or the in-bus vehicle network. Indeed, the existing in-vehicle communication systems are not secured by design. Thus, the in-vehicle network encounters many issues that don't meet the vehicular security

requirements.

CAN broadcast messages on the bus, where the receiver node selects the message based on IDs it is configured to receive. This enables attacks such as message interception (as illustrated in Fig 2.10) that violate the *confidentiality*. Also, lack of information about the sender in the message like signatures, an attacker can send arbitrary CAN frames to any node in the network and violate the *authenticity and non-repudiation*. The other case is, suppose an attacker manages to take over a gateway (See Fig 2.3) that forwards messages from one domain to another. In that case, the attacker can add, remove or change any data that the relayed message carries, meaning the violation of *integrity*. Besides this, since an attacker can send any data on the bus by sending high-priority messages (Section 2.1.3) or messages with error flags can cause nodes to stop responding causing a Denial of Service (DoS), thus affecting the *availability* of the system.

Our review analysis with the CAN bus's effects conclude that a cyber-attack, when observed at the CAN bus level, looks like a modification of CAN packets, i.e. CAN packet, is inserted, deleted or modified. Thus, yielding a categorization into three groups fabrication, suspension, and masquerades attack.

- **Fabrication Attack (Injection):** Fabrication attack is carried out by fabricating messages with a forged ID on a corrupted ECU and inserting them on the network, resulting in conflicting the ECU supposed to receive this ID Liu u. a. (2017); Choi u. a. (2018); Choi u. a. (2016). Fabricated messages are inserted with a higher frequency than the original message, and the original message is not taken into account. The fabricated messages can also be inserted either immediately before, or immediately after the occurrence of the original message.

- **suspension attack (Bus-off attack):** This attack consists of stoping the ECU from sending messages. When attackers continually send bits both in the identifier field and in other fields (See Fig 2.7), which causes the ECUs transmit error counter (TEC) then be incremented. When the TEC value is greater than 255, the corresponding ECU has to shut down Choi u. a. (2018).An other way is when attackers continually send high priority messages that block legitimate low priority messages (DoS attack) Liu u. a. (2017)

- **masquerading attack :** An attacker masquerades as a legitimate node. Liu u. a. (2017); Choi u. a. (2018); Tomlinson u. a. (2018a) identify two CAN vulnerabilities that facilitate masquerading attacks. First, CAN frame are not encrypted and can be studied by attackers to locate system entry points. Second, CAN does not support message authentication. The receiver of messages has no information about the source's validity, meaning that illegitimate messages are captured without being detected. This allows other action for the attacker like Replay attack, consisting of continually resend a valid message to prevent the vehicle real-time functioning Liu u. a. (2017); Nowdehi u. a. (2017); Mundhenk u. a. (2015); Tomlinson u. a. (2018b).

By reviewing the knowledge about the potential security vulnerabilities, threats and attacks, the research community had proposed many defence mechanisms to secure the in-vehicle architecture network. Thing und Wu (2016) proposed three main categories for in-vehicles defence: preventive defence, passive defence, active defence.

(A) **Preventive defence :** In preventive defence, securing the systems mainly focuses on protection measures to defend and attempt to stop an attack from happening or make them useless for the attacker. To this end, many methods had been proposed to rectify the lack of secure communication for the in-vehicle network. Mundhenk u. a. (2015); Kang u. a. (2017); Tashiro u. a. (2017) suggest a lightweight authentication using asymmetric cryptography, one-way hash chain and sending a partial MAC in each frame. Luo und Hou (2019) proposed Automotive Gateway Firewall.

(B) **Passive defence :** Attackers who can intrude on the in-vehicle system and bypass preventive defences can harm or cause damages. Thus, the passive defence should provide another layer of protection against the adversaries when the attack is happening. Intrusion Detection Systems (IDSs) are one of those defence mechanisms, different models of Intrusion Detection System (IDS) for in-vehicle network security have been proposed and tested under computational simulation scenarios Choi u. a. (2016); Choi u. a. (2018); Groza und Murvay (2018); Vuong u. a. (2015); Kang und Kang (2016). Anti-malware, also as a passive defence solutions for in-vehicle network should be capable of defending from harmful software that attempt to infiltrate the system. As malwares against vehicle are still in its early stage, there may not be a high number of malwares database available. Nonetheless, signature based detection can be put in place, by first considering these malwares Zhang u. a. (2014).

(C) **Active defence :** Countering advanced and determined adversaries would require an active approach to security. Using Continuous Security Monitoring critical components and interfaces. Besides this, Vehicles are considered vital systems, networks, and critical infrastructure that continue to evolve at a fast pace. Therefore, it becomes necessary to design and deploy defence measure such that they are themselves, moving targets. Adaptive reconfiguration of attack targets and deception tactics can be employed to enable better control and flip the balance during an attack. Also, detection models should also evolve through self-learning during their operation lifecycle to adapt to detect new forms of aggression.

This section provides an overview of in-vehicle network security characteristics, discussing its merits and shortcomings, ranging from intrusions and threats to specific CAN bus attacks. We point about several existing research work proposal that treats many security issues subject with recommendations for addressing and improving connected cars cybersecurity. Many challenges encounter developing a secured connected vehicle, due to its external connectivity and limited computational resources, lack of attack and threat database and the critical risk for passengers, including their lives. Network security solutions and situational awareness tools are essentially defensive strategies, and even the best defences can fail. Hence it is necessary to have a fallback solution that would prevent the vehicle from worst scenarios (taking unreasonable decisions that would endanger the vehicle's occupants). A fallback solution would prevent such an attack because the vehicle would know how to differentiate between normal and derivative state of it inners network system. Hence the Intrusion Detection system is a fundamental component

of the general in-vehicle security system.

In the following sections, we will review the IDS definition and its taxonomy related to the in-vehicle network; we will define the multiple constraint and challenges that bind the development of practical IDS for the in-vehicle network system. This knowledge will provide the basics challenges and questions that will serve as problems that we will tackle in our next chapters.

2.4 In-vehicle Intrusion Detection System

As discussed above, and based on researcher and experts in cybersecurity, connected systems in general, will never be absolutely secure Bellovin (2001). Besides this, finding and fixing the system's security flaws is not feasible for technical and economic reasons. In the case of vehicles, it demands the recall of all carsDenning (1987). We also discussed that the attackers might bypass standard security mechanism. Even if the attack is not active or successful, it is always useful to be aware of the intrusion attempts. Thus, developing a mechanism that can deal with threats while they are in action is valuable for any system's security, especially dynamic system like connected vehicles, where such a mechanism is independent of the system's functionalities and defences.

In 1985, Dorothy und Neumann (1985) was one of the earliest research work on Intrusion Detection System (IDS), he introduced IDS under the name *a real-time intrusion-detection expert system (IDES)*. Under the hypothesis of an intrusion or exploitation of a system's vulnerabilities involves a deviation or abnormal system usage patternsDenning (1987). The IDS task is to detect and identify those malicious activities by monitoring the system's activity independently from any particular subsystem, application or type of intrusion, thereby aiming for general-purpose abnormality detection. The IDS monitors and analyses the system's events and employs modelling techniques to recognise intrusive behaviour in a system. IDS have been classified in the literature using different taxonomies. Our work will present a brief taxonomy grouping the leading dimension criteria to build and in-vehicle IDS. depending on the different context and fields Bijone (2016b); Liao u. a. (2013); Veeramrddy u. a. (2011); Garc2a-Teodoro u. a. (2009); Buczak und Guven (2016); Lazarevic u. a. (2005)

To build such a system, many dimensions need to be considered. Indeed, the assets that might have a value or power in their context require specific monitoring sources and the use of tailored methods. Also, infrastructure architecture may require a particular computation location with its resources constraint. The evaluation of the IDS output demand informative metrics to assess its performances, based on this output, the usage and experts decision-making needs to be considered. We describe below a taxonomy of the different part and their relation in building an IDS.

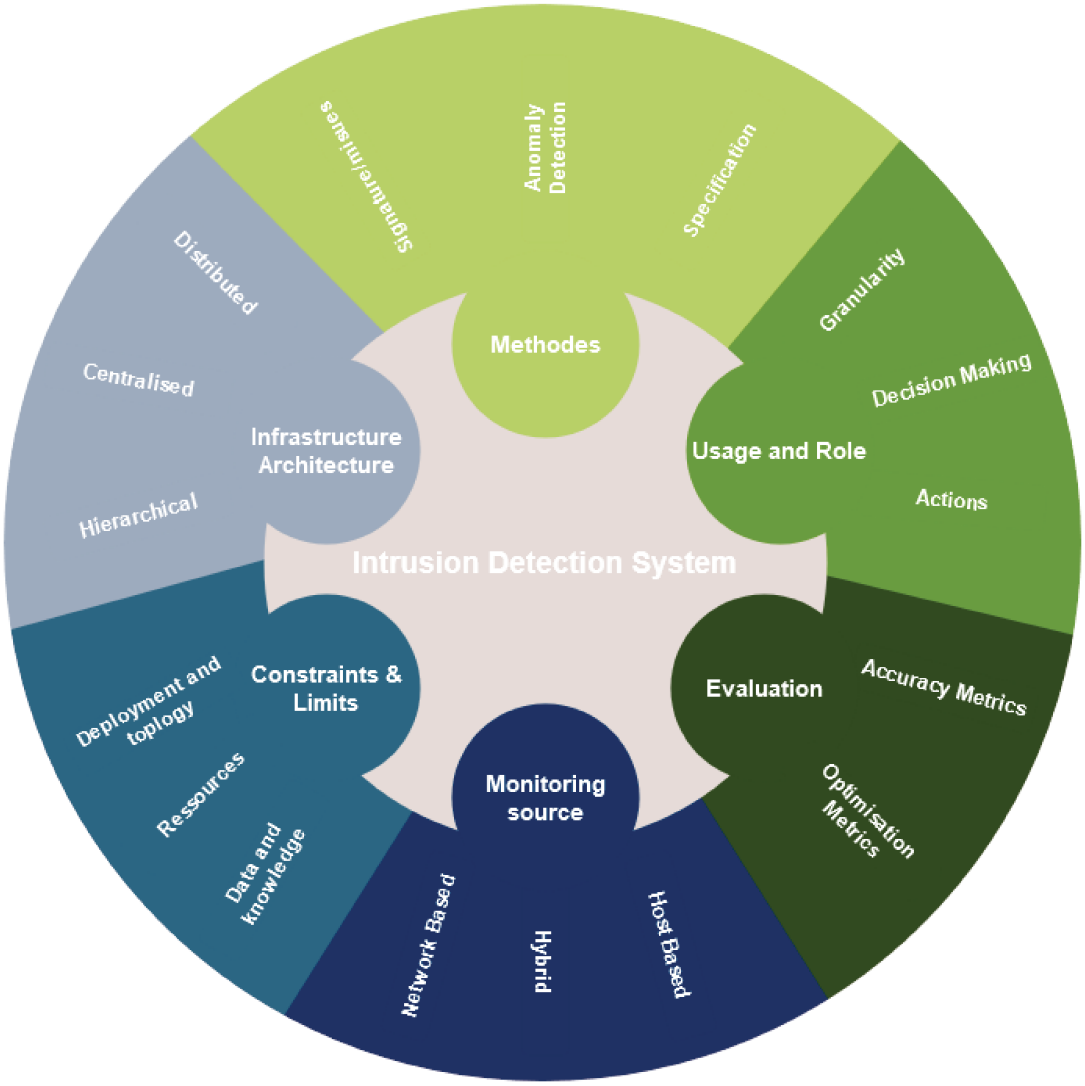


Figure 2.14: Different dimensions of In-vehicle Intrusion detection system.

2.4.1 A Brief Taxonomy

Architecture and deployment

The IDS performs its analysis based on local data provided by a set of independent probes deployed in different monitored distributed systems. These observation devices are responsible for monitoring a small, well-defined part of the entire system. The reported information either corresponds to a raw observation of activity (Local data) sent to a centralised method for a global analysis or results from a first low-level analysis of distributed or a hierarchical method that identifies the local state (suspicious local behaviour or normal behaviour). Depending on the configuration, the method used for the detection needs to fit the architecture deployment. In both cases, the IDS usually sends the results to a global analysis tool (Security Information and Event Management (SIEM)) responsible for aggregating the different security mechanisms deployed on the system for further mitigation and analysis using some aggregation and correlation engine.

Monitoring source

Whether a probe monitors a particular machine's activity or the activity at a specific point in the communication network, IDSs are classified into three main classes: HIDS (Host-based Intrusion Detection System) NIDS (Network-based Intrusion Detection System), and Hybrid IDS. A host-based IDS monitors the local behaviour on a single operating system host, generally analysing system performance, program process, and the operating system logs Debar u. a. (1999); Guha und Kachirski (2003); Kruegel (2004). Network-based IDS monitors network traffic (the metadata and content of the packets sent into the network) to observe network activity and detect if it may be a part of an attack process. The hybrid IDS monitors both host and network sources jointly to capture their correlated information since both sources complement each other, mainly concerning achieving a broader coverage for detection.

Methods

The Signature-based (misuse/Knowledge) Approach: the signature-based approach relies on the apriori knowledge of some possible attacks and attempts to encode this knowledge as rules to define malicious traffic/activity patterns to compare to current samples. If a match occurs, an alarm is raised. In that case, the signature-based approach has the advantage of causing very few false positives if the attack's correlation rules are sufficiently accurate. In the other hand, this method can only detect a known attack. A new attack or even a new instance of known attack (Homomorphe attack) are not detected (0-day exploit). Moreover, detecting a known attack occurs if the probes are well placed to cover all the system data required and correctly configured Kruegel und Toth (2003); Krügel und Toth (2002). Recently, misuses detection incorporates techniques allowing more flexibility to deal against the variations of attacks, using *Machine Learning* and augment the rule-based by using *Fuzzy Logic*.

The anomaly-based (behaviour) approach: This method start first by defining the pattern of normality, the method analyses whether the current activity samples deviate from the established model of normality and if so, an alarm is raised. This approach is more exposed to false alarms. Indeed, The network traffic is susceptible to perturbation and sometimes to evolutions. The anomaly-based approach Tsai u. a. (2009); Wu und Banzhaf (2010); Hamed u. a. (2018); Buczak und Guven (2016) interprets those events as anomalous, even though they are intrinsic network behaviour. One of the benefits of anomaly detection is detecting new attacks since the system is modelled according to normal behaviour. Anomaly-based IDS challenge is keeping up to date with environmental changes by retraining, or continuous updating is required to avoid an increase in false alarms, referred to as behavioural drift Kruegel (2004). Many works have proposed the hybridisation of techniques to improve the detection rates by combining both misuses and anomaly detection for IDS since they found that different methods performed better on different intrusion Ozcelik u. a. (2017); Garg und Maheshwari (2016); Bostani und Sheikhan (2017).

The Specification-based approach: Specification-based intrusion and attack detection systems generate speci-

fications from the intended target features, e.g., protocol state machines and security-related behaviour. The use of security specifications to detect malicious activity has been proposed and investigated in several contexts Larson u. a. (2008); Ko u. a. (1994); Majeed und Altaf (2019); Cheung u. a. (2006).

Usage and role

A real-time intrusion detection tools analyse the activity of the system to be protected and enable defensive actions. Audit data is examined as soon as it is produced based on different granularities in time. The advantage of this approach is that system activities can be analysed timely. Thus, a proper response can be issued when an attack is detected, which defines the IDS (responding with a passive notification or active reaction). However, real-time collection and analysis of audit data may introduce significant overhead in the communication network. offline tools are run offline at specific intervals. They analyse a snapshot of the system state and produce an evaluation of the security of that state. They do not provide any protection between two consecutive runs. Therefore, in case of a successful attack, they can be used only for postmortem analysis or to improve and update the IDS dynamic model. However, they may perform a more thorough investigation by occasionally running without an unacceptable impact on the monitored system performance. As explained above, IDS can be distributed following different location in the system. To this end, a decision about the system state can be made collaborative between separate local and complement each other's coverage analysis or an independent fashion analysis specific for each probe.

Evaluation

For an IDS to be considered effective, high detection rate and low false-positive rate are important aspects to consider. Multiple metrics could be used for an IDS evaluation Hodo u. a. (2017). These metrics are discussed subsequently showing the significance and purpose of each. It is essential to mention that an evaluation metric doesn't reflect an IDS performance depending only on the detection rate. Indeed, we need other important evaluation factors, including the transparency and safety of the overall system, memory requirements, power consumption, CPU consumption and throughput via communication overhead, should be considered. Specifically, these metrics are essential for IDSs running on different hardware or specific settings such as high-speed networks, or hardware with limited resources. The result's clarity and explainability are also important aspects of an IDS, which helps the forensic analysis and speed-up the decision-making process.

The term accuracy is defined in the following :

- True Positive (TP): Number of intrusions correctly detected
- True Negative (TN): Number of non-intrusions correctly detected
- False Positive (FP): Number of non-intrusions incorrectly detected

- False Negative (FN): Number of intrusions incorrectly detected

In the IDS terms, false-positive refers to security systems incorrectly detecting legitimate requests as misbehaviour or security breaches. In other words, the IDS detect something that not supposed to. Alternatively, IDS is prone to false-negatives where the system fails to detect something it should. False-negative imply missing detection of attacks that will not be mitigated and give a false sense of security. From the IDS perspective accuracy, many metrics are used:

$$\textit{OverallAccuracy} = \frac{\mathbf{TP} + \mathbf{TN}}{\mathbf{TP} + \mathbf{TN} + \mathbf{FP} + \mathbf{FN}} \tag{2.1}$$

Equation 2.1 provides the overall accuracy. It returns the probability that an item is correctly classified by the IDS.

$$\textit{Precision} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FP}} \tag{2.2}$$

Equation 2.2 provides the percentage of positively classified incidents that are truly positive.

$$\textit{recall} = \frac{\mathbf{TP}}{\mathbf{TP} + \mathbf{FN}} \tag{2.3}$$

Equation 2.3 provides the percentage of positively classified incidents based on the model positive prediction and it is also called *Sensitivity*.

To visualize the performance of an IDS, The trade-off between recall (true positive rate) and precision (true negative rate), AUC (Area Under The Curve) is used. Also, Equation 2.4 represents the harmonic mean of precision and recall. F1 is better suited to describe the performance of an IDS, especially when dealing with imbalanced classes.

$$\textit{F1} = \frac{\mathbf{2TP}}{\mathbf{2TP} + \mathbf{FP} + \mathbf{FN}} \tag{2.4}$$

in-vehicle network Constraints and Limits

Building IDS limitation and constraints depend on several dimensions; the context and topology of the studied system, its architecture, Deployment to its evaluation and the used methodology. The in-vehicle intrusion detection

system can be considered as an embedded system, a computing system with a combination of hardware and software designed to perform one or a few dedicated functions with some vehicular domain specifications. We separate those concerns into Four elements:

1) *The data and knowledge*: One of the general questions is about the quality of the data and the knowledge experience about attacks and threats, compared to the IT system, that has more history and experience dealing with a different type of attack, the in-vehicle network was a closed system, and the threats exposed now are mostly subject of research efforts. What kind of data attack systems need to observe, while the in-vehicle system has various data sources. The more data can be monitored and obtained for evaluation, the better the overall picture of the system's current situation. However, the more information needs to be observed, gathered and evaluated, the more complex and costly the development and analysis process becomes.

2) *Ressources and optimisation*: An attack detection system needs to fulfil real-time performance requirements Kargl u. a. (2008); Carsten u. a. (2015); Robinson-Mallett und Hansack (2015); Lee u. a. (2014) with the capacity of reducing the false-positive while being accurate. Especially attacks that target the vehicle's safety, e.g., by sending false messages to the brakes, engine, etc. can only be tackled if this requirement is fulfilled. However, the automotive environment is a network of embedded systems, including highly specialised and cost-optimised components, which offer only limited computational power but are designed to work reliably under very different physical conditions Kargl u. a. (2008); Müter u. a. (2010).

3) *Deployment and toplogy*: As explained above, the in-vehicle system also require a particular type of Deployment to fit the topology and the architecture of the in-vehicle network, this includes a strategy to handle the different data sources and how to implement the IDS model while respecting the two previous concerns.

4) *Industrial needs*: The in-vehicle network architecture is fixed after the client's purchase. The IDS automation needs to consider the cost of the inclusion of the ECUs that will be part of the IDS Hindy u. a. (2020); Zhou u. a. (2015). Also, a practical solution is a solution that takes into consideration the industrial needs in terms of usage and role (See Fig 2.14).

2.5 Conclusion

This chapter has provided the context of this research study concerning in-vehicle network security. We present a consistent background development of security from IT to the recent advanced vehicles architectures. We briefly discussed the several defence classes and solution while highlighting a focus on Intrusion Detection System in general. We extend the Intrusion detection system survey to a taxonomy specific to in-vehicle network demand and constraints.

Chapter 3

The anomaly detection problem : state-of-the-art

Vehicles have a very long life span and are in use for decades in different conditions and locations. As discussed above In Chapter 2, to protect the car over this long period, only preventive measures are insufficient, and the vehicles security system has to work autonomously without a necessity for user interaction. To this end, IDSs are a good candidate to complete the security architecture of vehicles, with the capability to monitor the traffic on the vehicular networks and continually evaluate abnormal events to classify them as an attack or not. If appropriate, an alarm is raised as soon as a threat is detected, countermeasures to respond to attacks are considered, if reactive components have been integrated into the car's security system. This thesis presents an IDS scheme for in-vehicle networks that respect automotive networks' typical characteristics, and its protocols like the Controller Area Network (CAN). One central question is how exactly the IDS should identify attacks for the in-vehicle network, which method detection is suitable for the automotive area. Signature-based and Specification-based approaches and detection Dupont u. a. (2019); Ji u. a. (2018); Olufowobi u. a. (2020); Larson u. a. (2008); Dupont u. a. (2019) promises a low false-positive rate, which is important as numerous false alerts could question the usability of the entire concept in the vehicle and may negatively affect the driver's awareness. However, as cited above, one major constraint of that method is dealing with unknown attacks and the lack of attack database specific for in-vehicle systems and need a rule for every single intrusion and variation thereof. For specification-based, handling the different components of the cars made with various providers makes it not practical to define explicitly the system normal behaviour limits.

Anomaly detection promises to detect attacks, including novel attack patterns that result in a system state which differs from the normal specification. However, in the past, anomaly detection systems were typically prone to high false-positive rates, and the specification of the system's normal behaviour has turned out to be challenging.

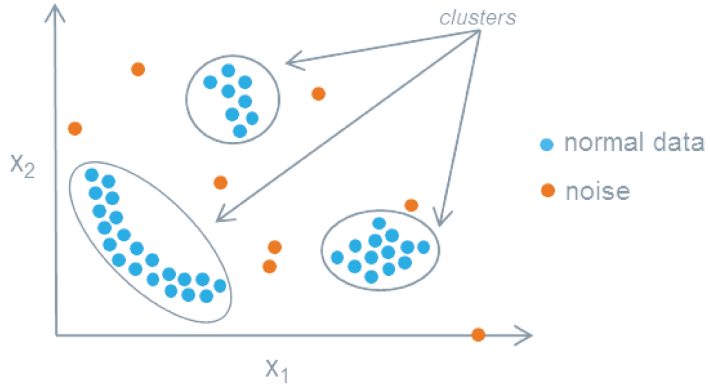


Figure 3.1: Illustration of Anomalies VS Novelty in two-dimensional data-set.

Nevertheless, with the advance in machine learning approaches and the data-driven approaches, we suppose the vehicular networks' normal behaviour can successfully be defined and adopted. In that case, we consider anomaly detection to be the more promising approach to start within the automotive domain as unknown attacks may be detected. A challenge to either approach is updating the system. For rule-based systems, this involves adding new rules and potentially updating old rules. The drawback of rule-based systems is that the knowledge base of rules may grow very large with time and does not scale well Jiang und Cybenko (2004). For some machine learning techniques, updating may involve complete re-training, including gathering data concerning new intrusions or the system's new normal states. Some methods can learn continuously online, and in our case, we consider an update as an offline process because the danger is that intrusive behaviour can also be learned.

It is clear that either approach exhibits specific pros and cons, and that neither method can be said to be 'better' than the other, but complement each other. In the future, hybrid practices can be advantageous when mature experience related to vehicle attacks and threats are available.

3.1 Problem statement

When analyzing monitored network datasets, a common need determines which instances stand out as being dissimilar to all others. Such instances are known as anomalies. The goal of anomaly detection (also known as outlier detection or novelty detection in some works) is to determine all such instances in a data-driven fashion Chandola u. a. (2009); Chandola und Kumar (2007). Errors in the data can cause anomalies. In many cases, it indicates a new and previously unknown pattern in the data. An outlier is an observation that deviates so significantly from other observations to stimulate suspicion that a different mechanism generated it Hawkins (1980) due to several reasons such as malicious actions, system failures, intentional fraud or cyber-attacks. Anomalies also referred to as abnormalities, outliers or misbehaviour in the literature Aggarwal (2013), are considered from a data perspective as points or regions which are located further away from bulk areas consisting a majority of observations considered

as normal data instance region (See Fig 3.1). Novelty is not always considered an anomaly; it is an unobserved pattern. They are used to set a novel score for the threshold. The points that significantly deviate from this decision threshold may be considered anomalies or outliers Miljković (2010); Pimentel u. a. (2014). Novelty Detection and Anomaly Detection have a slightly different meaning, where the difference resides in how far the deviation is from a normal pattern (See Fig 3.2).

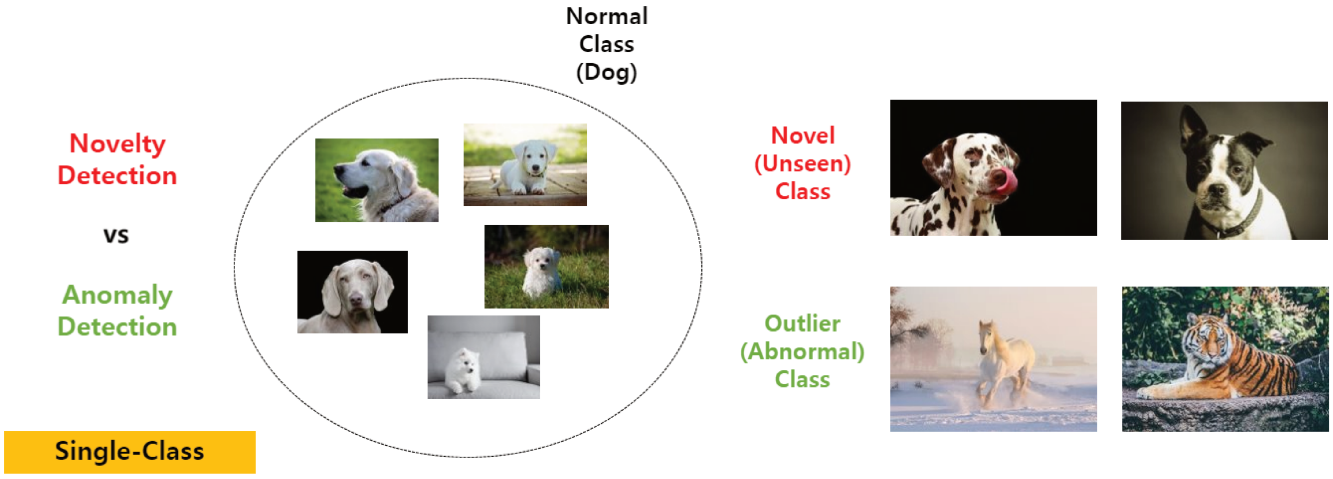


Figure 3.2: Illustration of Anomaly VS Novelty in the image data set.

Chandola u. a. (2009); Chandola und Kumar (2007) classifies Anomalies into three types: point anomalies, contextual anomalies and collective anomalies.

3.1.1 Anomalies Types

(A) **Point anomalies:** Often represent an irregularity or deviation that happens randomly and may have no particular interpretation; the point deviates significantly from the rest of the data points. This point represents a single occurrence of an event that does not fit the ordinary (See Fig 3.1).

(B) **Contextual Anomalies:** A contextual anomaly known as the conditional anomaly is a data instance that could be considered anomalous in some specific context (context-sensitive Anomalies) Song u. a. (2007). Contextual anomaly is identified by considering both contextual and behavioural features. The contextual features, normally used are time and space. Simultaneously, the behavioural features may be the pattern of spending money, system log events or any feature used to describe the normal behaviour. For example, the speed of a car can be considered abnormal, given the road traffic state.

(C) **Collective or Group Anomalies:** Collections of individual data points are known as collective or group anomalies. Each point appears as normal data instances while observed in a group exhibit unusual characteristics. Group anomaly detection is irregular group distributions (e.g., irregular mixtures of image pixels

are detected) Chalapathy u. a. (2019).

3.1.2 Anomaly detection models output

Anomaly score describes how far each data point is from the normal pattern. According to the anomaly score, the subject matter expert will select a domain-specific threshold (commonly known as decision score) to identify the anomalies. In general, decision scores reveal more information than binary labels depending on the type of methods used. For instance, in OC-SVM, the decision score measures the distance of data point from the centre of the sphere; the data points that are farther away from the centre are considered anomalous. The anomaly score is used after setting a threshold to automate the model's decision to assign a category label as normal or anomalous to each data instance. AD model can also return a multi-class label (normal/anomalous-A/anomalous-B/...).

3.1.3 Anomaly detection challenges

The main advantage of systems based on this approach is that, in theory, they can detect previously unknown attacks. On the other hand, anomaly-based systems are prone to many challenges:

- The amount of time required to train the system the normal behaviour might be long.
- There is no universal procedure to model normal data. The concept of normality is still subjective and difficult to validate.
- The anomalous examples in datasets are not often available, and when they are the classes (normal/anomalies) are generally imbalanced. Besides, the data may contain noise which makes the distinction between normal and abnormal data more difficult.
- The monitored environment's behaviour might change during a period, requiring the system to updated to avoid false alarms.
- Performance of traditional algorithms in detecting outliers is sub-optimal on the image (e.g. images or distributed topology network) and sequence datasets since it fails to capture complex structures in the data.
- Need for large-scale anomaly detection: As the volume of data increases, it becomes nearly impossible for the traditional methods to scale to such large scale data to find outliers.
- Solve the problem end-to-end taking raw input data in domains with automatic feature learning and eliminates the manual developing of features by domain experts, especially in a closed system like vehicles systems.
- The boundary between normal and anomalous behaviour is often not precisely defined in several data domains and is continually evolving. This lack of well-defined representative normal boundary poses challenges. For instance, it may be possible to perform an attack within the boundaries of normality.

3.2 Machine learning & Computational intelligence for AD

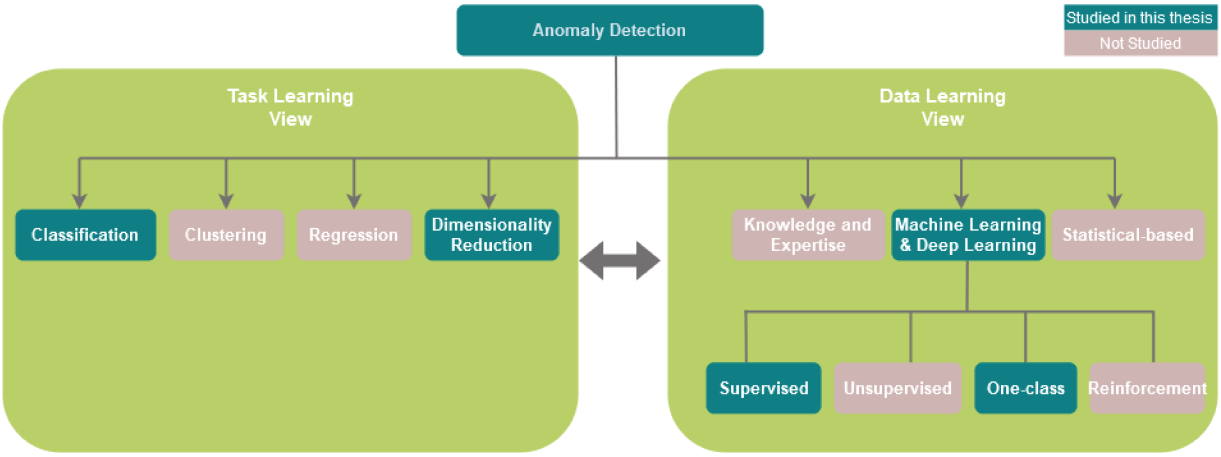


Figure 3.3: Research area used for anomaly detection

Given large content of the literature on anomaly detection techniques, these approaches differ in many aspects, such as the applicability scope, the execution mode (offline/online), and domain knowledge requirements. An essential element of anomaly detection is the nature of the data input. Input represents the record, points or pattern of events with many characteristics and types, from univariate points to time series (sequential data), spatial and graph data, images and videos, and multi-modal inputs combining different types. Many research fields proposed and explore many techniques and methods (See Fig 3.3) to cope with anomaly detection process and the different natures of data.

We have grouped works in a few common threads: a rule-based, Machine learning and Deep learning and statistical-based. We explain the different techniques and enumerate their pros and cons based on literature applications. We not that our primary focus is on applying Deep Learning for in-vehicle IDS based on anomaly detection. Nonetheless, we review the standard and classical approaches of the literature that tackled anomaly detection and we review the the application of Deep learning techniques for anomaly detection below in (Section 3.3).

3.2.1 Knowledge and Expertise

It's an extension of *specification-based* with using Information theory, decision-tree or Fuzzy theory to enhance the actual manually developed characteristics of legitimate CAN network behaviours Dupont u. a. (2019); Weber u. a. (2018); Linda u. a. (2011). Those approaches assumes that there exists a set of rules that allows identifying anomalies in a deterministic way. This approach requires labelled data. It learns rules that capture the normal behaviour of a system. A test instance that is not covered by any such rule is considered an anomaly. Many techniques allow rule induction, such as decision trees and association rules Brauckhoff u. a. (2009); Chandola u. a. (2009); Yang u. a. (2019). In Marchetti und Stabili (2017), propose a model of the normal behaviour of a CAN

network-based on a particular feature: recurring patterns within the sequence of message IDs observed in the CAN Bus by modelling the CAN bus's normal behaviour in the form of transition matrix. This data structure identifies all the legit transitions between the message IDs of two consecutive CAN messages. This model will then be used as a reference to identify anomalies in the CAN traffic.

3.2.2 Statistical based Techniques

In statistical-based techniques Chandola u. a. (2009), the network traffic activity is captured, and a profile representing its stochastic behaviour is created. This profile is based on metrics such as the traffic rate, the number of packets for each protocol, the rate of connections, the number of different IP addresses, etc. Two datasets of network traffic are considered during the anomaly detection process: one corresponds to the currently observed profile over time, and the other is for the previously trained statistical profile. As the network events occur, the current profile is determined, and an anomaly score is estimated by comparing the two behaviours. The score indicates the degree of irregularity typically for a specific event, such that the intrusion detection system will flag the occurrence of an anomaly when the score surpasses a certain threshold. In other words, anomaly detection methods in these categories are based on estimating the probability densities of the data using statistical models and assuming that normal data will fall in high probability regions. In contrast, anomalies will fall in low probability ones Chandola u. a. (2009); Pimentel u. a. (2014); Basora u. a. (2019). In Chandola u. a. (2009); Chandola und Kumar (2007); Hawkins (1980); Aggarwal (2013); Miljković (2010); Pimentel u. a. (2014); Song u. a. (2007); Basora u. a. (2019) categorise further methods both parametric and non-parametric techniques with a multitude of applications. As our work focuses more on Deep Learning approaches, we will review a few techniques that have been widely used.

- **chi-square test statistic**: Ye und Chen (2001) use chi-square theory for anomaly detection, according to this technique, a profile of normal events in an information system is created, and the idea in this approach is to detect the large departure of events from the normal profile as anomalous or intrusion. To this end, a distance measure based on the *chi-square* test statistic is developed as :

$$\chi^2 = \sum_{i=1}^{n} \frac{(X_i - E_i)^2}{E_i} \tag{3.1}$$

  Where X_i is the observed value of the ith variable, Ei is the expected value of the ith variable and n the number of variables.

  The X^2 has a low value when observing the variables is near the expected value. So, following the $\mu \pm 3\sigma$ rule, an anomaly is raised when X^2 value is higher than $X^2 + 3S_x^2$. In Krügel u. a. (2002), the author proposed

a statistical processing unit for detecting anomalous network traffic to detect rare events like R2L or U2R (See Section 2.2.2). The metric aim to search identical characteristics of different service requests. The anomaly score is calculated based on the type and the length of the request and the payload distribution. In Moore u. a. (2017) exploit the regularity in the timing of CAN communication. For each CAN ID, the NIDS stores the time differences between two successive messages and computes the mean arrival time. It will also register the maximum time difference from the mean, which will be used to define a threshold. An alert will be raised if the time between two packets differs from the expected time by more than the maximum time difference plus 15% of the mean.

- **Mixture model**: In the category of parametric techniques Chandola u. a. (2009) assume that the normal data is generated by a mixture of parametric distributions with parameters and probability density function $f(x, \Theta)$, where x is an observation. Such techniques use a mixture of parametric statistical distributions to model the data. We find Gaussian Mixture Models (GMM) based on the assumption that the data is generated from a weighted mixture of Gaussian distributions. We find two subcategories. The first subcategory of techniques models the normal instances and anomalies as separate parametric distributions. The testing phase involves determining which distribution the test instance belongs Chandola u. a. (2009); Chandola und Kumar (2007); ABRAHAM und BOX (1979). In contrast, the second subcategory of techniques models only the normal instances as a mixture of parametric distributions. A test instance that does not belong to any learned models is declared an anomaly Chandola u. a. (2009); Agarwal (2006).

  In Eskin (2000), the author proposed an IDS based on mixture models. The approach estimates a probability distribution over the data and applies a statistical test to detect the anomalies in *UNIX system*. The set of system calls having a probability of $1 - \lambda$ is a legitimate use of the system, and the intrusions have the probability of λ. The two probability distributions which generate the data are called the majority (M), and anomalous (A) distributions Ahmed u. a. (2016b). In Liang u. a. (2020), the author proposes a novel filter model based on a hidden generalized mixture transition distribution model (HgMTD) in VANETs, which can quickly filter the messages from neighbouring vehicles. It adopts a well-known multi-objective optimization (NSGA-II) algorithm combined with an expectation-maximization (EM) algorithm to forecast neighbouring vehicles' future states and then filter out malicious messages by monitoring the change of the state pattern of each neighbouring vehicle.

  In Hamada u. a. (2018), The proposed method learns the Probability Density Function (PDF) of the Gaussian mixture model, from the reception cycle periods of the monitored CAN-ID using the sequentially discounting expectation and maximization(SDEM) algorithm. These reception cycle periods are calculated from the arrival time of each CAN frame of the monitored CAN-ID. The learner continuously estimates each PDF from the reception cycle periods in each slot. The anomaly detector assesses each CAN frame using the PDF that was

estimated in the previous slot. A threshold is used to assess the frame in the anomaly detector.

- **Classical regression with statistics:**

  The regression model-based anomaly detection is a subcategory of the parametric techniques identified inChandola u. a. (2009). This approach is based on the principle of forecasting. A regression model is first fitted on the training data. The resulting model is then used on test sequences to compute the residuals (the difference between the predicted value and the real value). The residuals determine the anomaly scores. In classical statistic approaches for the regression task, we can include anomaly detection techniques based on traditional time series forecasting models such as Vector Auto-Regressive (VAR) Melnyk u. a. (2016), and Autoregressive Integrated Moving Average (ARIMA) Bianco u. a. (2001); CHEN u. a. (2005); Zhu und Sastry (2011). In Tomlinson u. a. (2018c) two methods, ARIMA and Z-score, were using broadcast time-intervals to detect injected packets in the highest priority and regular broadcast CAN packets. The author showed similar results to the supervised method using prior average times.

Statistical approaches have several advantages to enhance Anomaly Detection process. Firstly, they do not require prior knowledge about the target system's normal activity; instead, they can learn the system's expected behaviour from observations. Secondly, statistical methods can accurately notify malicious activities occurring over long periods with tasks like regression. However, many drawbacks pointed out Garc3a-Teodoro u. a. (2009); Basora u. a. (2019); Chandola u. a. (2009). First, an attacker so that the network traffic generated during the attack is considered normal. Second, setting the different parameters/metrics' values is a difficult task, especially because the balance between false positives and false negatives is affected. Moreover, a statistical distribution per variable is assumed, but not all behaviours can be modelled using stochastic methods. Furthermore, most of these schemes rely on the assumption of a quasi-stationary process, which is not always realistic. And in more large scale computation, classical methods have their limitation.

3.2.3 Classical Machine learning approaches

Due to the thesis scope, some techniques and methods are not considered in this brief ML review. How ever, a broad range of ML and pattern matching methods are reviewed in Chandola u. a. (2009); Chandola und Kumar (2007); Hawkins (1980); Song u. a. (2007); Wu und Yen (2009). We focus on the main methods that have been applied to Anomaly detection and in-vehicle IDS. In Fig 3.3 we defined the main categories of Machine Learning tasks that we will review, from which we will mention the appropriate algorithm and techniques.

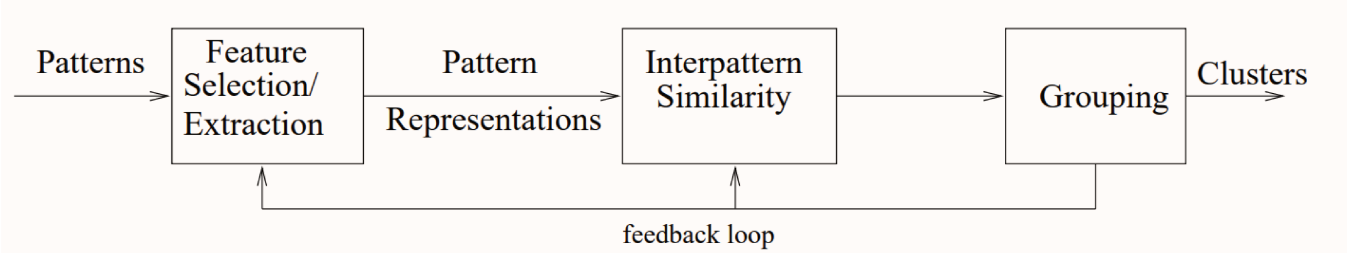


Figure 3.4: Illustration of The Clustering process flow

Anomaly Detection as Clustering Problem

Clustering refers to unsupervised learning algorithms which do not require pre-labelled data to extract rules for grouping similar data instances Jain u. a. (1999); Govaert und Nadif (2014). In Fig 3.4 we show the general process of clustering. Many techniques and variants of clustering have been used to perform anomaly detection Chandola u. a. (2009); Chandola und Kumar (2007); Kiss u. a. (2014); Ahmed u. a. (2016a). During the training phase, the data points are grouped into clusters over many iterations until convergence or attaining predefined criteria (maximum number of iterations and the used model parameters). In the test phase, a new data point is assigned to the closest clusters based on the distance/similarity measure used in training based on the type of data. The anomaly detection as a clustering problem relies on the assumption that any subsequent data that do not fit well with existing clusters of normal data are considered anomalies. Also, when a cluster contains both normal and abnormal data, it has been found that the normal data lie close to the nearest clusters centroid, but anomalies are far away from centroids. For a group of anomalies, the smaller and sparser can be considered anomalous and the thicker normal. Instances belonging to clusters' sizes or densities below a threshold are considered anomalous Ahmed und Mahmood (2013). Many approaches algorithms have been used to achieve the clustering phase, *K-means* is used in many works for clustering-based anomaly detection. Once clustering is achieved, an instance is classified as normal if it is closer to the normal cluster and anomalous otherwise. A threshold is defined, representing the distance between the centroid and the instance Münz u. a. (2007). In Syarif u. a. (2012), the authors investigated the performances of various clustering algorithms when applied for Network anomaly detection. They used five different approaches, the *k-means, improved k-means, kmedoids, Expectation-Maximization (EM)* clustering, and distance-based anomaly detection algorithms. Advanced techniques for clustering also have been proposed for AD, like *blockclustering* and *Co-clustering* Govaert und Nadif (2008, 2014). According to Ahmed und Mahmood (2014); Papalexakis u. a. (2012), co-clustering is beneficial for detecting DoS attacks and collective anomalies in network IDS in general. Eskin u. a. (2002) propose a geometric framework for unsupervised anomaly detection. In this framework, data elements are mapped to a feature space which is typically a vector space. Anomalies are detected by determining which points lie in sparse regions of the feature space. We present two feature maps for mapping data elements to a feature space. The first map is a data-dependent normalization feature map applied to network connections. The second feature

map is a spectrum kernel which applied to system call traces. In Lotfi Shahreza u. a. (2011) *Neural Network* based technique was proposed, Self-Organizing Maps (SOMs) are among the most well-known, unsupervised neural network approaches to clustering, which are very efficient in handling large and high dimensional datasets. The SOM architecture is a feed-forward neural network with a single layer of neurons arranged into a rectangular array. When an input pattern is presented to the SOM, each neuron calculates how similar it is to its weights. The neuron whose weights are most similar (minimal distance d in input space) is declared the winner of the competition for the input pattern, and the weights of the winning neuron are strengthened to reflect the outcome. The winning neuron receives the most learning at any stage; with neighbours receiving less, they are further away from the winning neuron. The original Particle Swarm Optimization (PSO) is another algorithm discovered through simplified social model simulation, which is effective in nonlinear optimization problems and easy to implement. In Lotfi Shahreza u. a. (2011) study, the authors combine these two methods and introduce a new method for anomaly detection for intrusion detection.

In Barletta u. a. (2020) the authors propose an efficient and high-performing intrusion detection system based on an unsupervised Kohonen Self-Organizing Map (SOM) network, to detect attack messages sent on a Controller Area Network (CAN) bus. The SOM network found a wide range of intrusion detection applications because of its features of high detection rate, short training time, and high versatility. The authors extend the SOM network to intrusion detection on in-vehicle CAN buses. Many hybrid approaches were proposed to combine the SOM network with other clustering methods, such as the k-means algorithm, to improve the model's accuracy (See Fig 3.5).

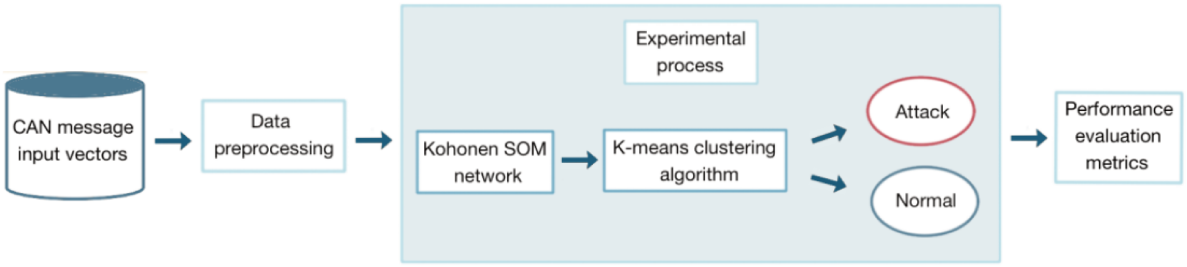


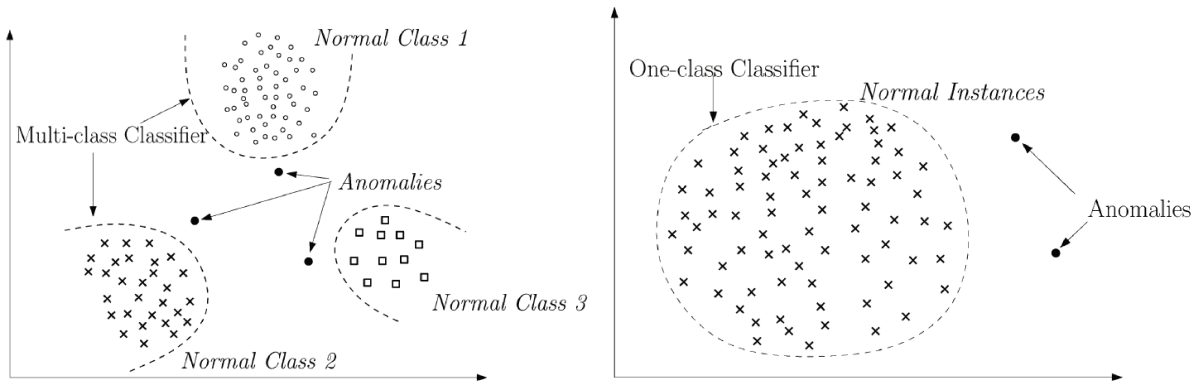
Figure 3.5: Experimentation process of In-vehicle Anomaly detection based on Clustering techniques Barletta u. a. (2020)

In summary, the clustering approach is very interesting in anomaly detection. It can operate in an unsupervised mode and does not require labelled data while allowing the detection of anomalies. The testing phase for new data is generally fast as it is a simple comparison with clusters predefined in the training phase. Such techniques can often be adapted to other complex data types by simply plugging in a clustering algorithm that can handle the particular data type (distance choice). However, depending on the algorithm, the training step may be very long as some algorithms have slow convergence issues. Besides, clustering-based techniques' performance relies on the effectiveness of clustering algorithms in capturing the cluster structure of normal instances based on the used

distance (or similarity measure). In high dimensional data, the distance significance may be altered by noise which may lead to poor performance. Also, Several clustering-based techniques are effective only when the anomalies do not form significant clusters among themselves; (based on the assumption that anomalies don't form clusters). The computational complexity for clustering the data is often a bottleneck when dealing with significant data volumes.

Anomaly Detection as Classification Problem

Classification-based techniques rely on experts extensive knowledge of the characteristics of network attacks. When a network expert provides details of the detection system's characteristics, an attack with a known pattern can be detected as soon as it is launched. This depends solely on the attack's signature as a system capable of detecting an attack only if its signature has been provided earlier by a network expert. With machine learning techniques, such a classification task can be performed using different techniques and algorithms. The goal is to learn a model *Classifer* represented with a function $f(.)$ from a set of labelled data instances (training) and then, classify a test instance into one of the classes using the learned model (test or inference). The learning process is about extracting the discriminative features characteristics that distinguish between normal and anomalous classes in the given feature space. Depending on the used technique, the goal is to take an input x and assigns it to one of the K classes (C_k). (In this work, we consider mainly the case of one and two classes, and investigate the extension to $K > 2$ classes in the chapter 4). Indeed, Based on the labels available for the training phase, classification-based anomaly detection techniques can be grouped into two broad categories: multi-class and one-class anomaly detection techniques Bishop (2006); Chandola u. a. (2009).



(a) Multi-class Anomaly Detection (b) One-class Anomaly Detection

Figure 3.6: Feature space representation of a Multi-class classifier for Anomaly Detection (a) and One-classe classifier Anomaly Detection (b) Chandola u. a. (2009)

- **Multi-class Anomaly Detection:**

Multi-class classification based anomaly detection techniques assume that the training data contains labelled instances belonging to multiple normal classes Chandola u. a. (2009). The method used teaches a classifier to distinguish between a normal class and anomalous examples (In most of the cases, anomalous instances do not form a class) in such a setting. Several variants of the basic neural network technique have been proposed using different types of neural networks Hawkins u. a. (2002); Williams u. a. (2002) (concerning those methods, an in depth review is presented below in the section 3.3). Also, much of previous research motivated the study of Anomaly detection, including Decision Trees (DT) that have been successfully applied to intrusion detection to discover known and unknown attacks. DT is a common classification method based on divide and conquers strategy. A DT comprises decision nodes and leaf nodes, and they represent a decision test over one of the features, and the result class, respectively Safavian und Landgrebe (1991). In Yang u. a. (2019), the authors propose selected ML algorithms based on a tree structure, including decision tree, random forest, extra trees, and XGBoost. In Kalkan und Sahingoz (2020) compared machine learning classifiers for the CAN security. As a result of the study, it has been observed that the decision tree-based ensemble learning models result in the best performance in the tested models. however, suffer from a number of drawbacks. One is that they cannot generalise to new attacks in the same manner as certain other machine learning approaches. They are not suitable for anomaly detection since homomorphic attacks and new attacks are a promising use of anomaly detection. Empirical findings also demonstrate that DTs are very sensitive to the training data and do not learn well from imbalanced data Chawla (2003); Gharibian und Ghorbani (2007).

Support Vector Machines (SVMs) have been applied increasingly to anomaly detection in the research in many application domains. SVM is one of the supervised learning models used classification problems. It uses each data's nearest training points to build one or more hyperplanes to classify high-dimensional data. It maximizes the margin between them as much as possible. The margin is defined as the perpendicular distance between the decision boundary and the closest of the data points. The decision boundary is chosen to be the one for which the margin is maximized Bishop (2006); Cortes und Vapnik (1995). An important property of support vector machines is that the model parameters' determination corresponds to a convex optimization problem, so any local solution is also a global optimum Schölkopf u. a. (2002).SVM classifiers have been used widely for intrusion detection to distinguish between normal and intrusive data Mukkamala u. a. (2002). And in Duan und Keerthi (2005) presented a common approach is to combine several two-class SVMs, and in Mukkamala u. a. (2005) applied this paradigm, to network-based intrusion detection, adopting the five class, thus requiring five SVMs. For each SVM, the training data is partitioned into two classes. One represents an original class, and the other class represents the remaining, e.g., Normal and all intrusions, or Probing and Normal and the other attacks. The combination technique adopted is a winner-takes-all strategy Duan und Keerthi (2005), in

which the SVM with the highest output value is taken as the final output.

Intrusion detection systems apply different SVM types to benefit from their capabilities, such as conducting multiclass classification. Moreover, in some approaches, to boost the ADS detection rate and accuracy, SVM parameters and its applied kernel functions are trained through using meta-heuristic algorithms. Also, to increase the detection rate, some of the ADS approaches have exploited the SVM with other classifiers such as decision trees, ANN, and na3ive byes. Furthermore, some of the outlined schemes have applied feature extraction methods or utilized meta-heuristic algorithms for feature selection. Finally, some other ADS approaches have tried to handle the imbalanced datasets, making the SVM training process faulty and biased. In Alshammari u. a. (2018), the authors propose an intrusion detection method for CAN bus IDS in vehicles. It detects DoS and the Fuzzy attacks which occur on CAN Bus. And in Tanksale (2019), propose a support vector machines based intrusion detection system that can detect anomalous behaviour with high accuracy. At the same time, they treated three overall types of anomalies (point anomaly, contextual anomaly, and collective anomaly). With respect to the CAN, each type of anomaly requires different message types and parameters to create the needed feature vectors. They grouped the CAN traffic features into groups based on the message's functionality as the SVM input feature vector.

- **One-class Anomaly Detection:**

  In a one-class problem, the training set only contains observations describing one class, and the classifier constructs a model that captures this class. For each new observation in the classification phase, the classifier decides if it falls inside this model or that it is an anomaly. A situation where one-class classification algorithms are practical is monitoring normal systems behaviour. In that case, Data describing normal behaviour can be gathered easily. Data relating to errors, attacks and security issues, cannot be collected easily and cost-effectively. Even with simulation, we cannot approach a sufficient distribution of anomalous examples that will encompass the behaviour of many types of attacks. We can collect CAN network traffic from cars under normal driving conditions, but traffic containing attacks is much more challenging to obtain. A one-class classifier is trained on the available Data and thus constructs a model describing the in-vehicle network behaviour under normal circumstances. The classifier decides for each new observation in the monitoring phase, whether it fits the class representing normal behaviour or anomaly. The Machine learning methods used for One-Class classification rely on the normal traffic activity profile that builds the knowledge base and consider activities deviate from the baseline profile as anomalous. The advantage lies in their capability to detect completely novel attacks, assuming that they exhibit ample deviations from the normal profile. One-class Support Vector Machine (OC-SVM), introduced Schölkopf u. a. (1999). aims to create a function that returns +1 in a region representing the training set's observations and -1 elsewhere. To achieve this, the same concepts apply as for the described binary problem, but now the distance between the origin and the hyperplane is maximised

(in the feature space F). Then, decision function $f(x)$ returns +1 for observations located on one side of the hyperplane and -1 for observations on the other side.

The process of automatically constructing models from data is not trivial, especially for intrusion detection problems. It's because intrusion detection faces problems, such as huge network traffic volumes, highly imbalanced data distribution, difficulty realizing decision boundaries between normal and abnormal behaviour, and a requirement for continuous adaptation to a constantly changing environment. This demonstrates a system that can only detect what it knows, and vulnerable to new attacks appearing in different versions. Additionally, the normal traffic not included in the knowledge base is considered as an attack. Thus, it generates unintentional false alarms. Continuous training is required for anomaly detection techniques to build a normal activity profile that is time-consuming and depends on the availability of a completely normal traffic instance. Furthermore, it is challenging to keep a normal profile up-to-date in today's dynamic and evolving network environment. Among a large pool of classification-based network anomaly detection techniques, we discussed four major classical methods and their related application to Intrusion detection for the in-vehicle network. The following section reviews the more recent and advanced technique for anomaly detection using Deep Learning.

3.3 Deep learning for Anomaly Detection

Deep learning for anomaly detection, deep anomaly detection for short, aim at learning feature representations or anomaly scores via neural networks for the sake of anomaly detection. Many deep anomaly detection methods have been introduced, demonstrating significantly better performance than conventional anomaly detection on addressing challenging detection problems in various real-world applications. Deep Learning methods enable end-to-end optimization of the whole anomaly detection pipeline, compared to traditional modelling(See Section 3.2) depends on feature engineering, which relies on expert and external knowledge to create features relevant to a given problem. Besides, they also allow the learning of representations specifically tailored for anomaly detection. These two capabilities are crucial to tackling the Anomaly Detection challenges, where traditional methods can fail. This subsequently results in more informed models and thus better recall-rate. For the anomaly explanation challenge, DL methods are black-box models; but they offer options to unify anomaly detection and explanation into single frameworks, resulting in explanation of the anomalies spotted by specific models (Deep Learning explainability is well-studied research, but not studied in this work). Deep Learning methods also excel at learning complex structures and relations from diverse data types, such as high-dimensional data, image data, video data, graph data, etc. This capability is important to address various challenges, multimodel-inputs, and an environment with many sensors like an autonomous vehicle. Further, they offer many practical and easy-to-use network architectures and frameworks to learn unified representations of heterogeneous data sources seamlessly. However, shallow meth-

ods for handling those complex data are generally substantially weaker and less adaptive than the Deep Learning techniques.

3.3.1 Deep Learning for Anomaly detection: Categorization and formulation

Deep neural networks leverage complex compositions of linear/non-linear functions that a computational graph can represent to learn expressive representations Goodfellow u. a. (2016). The two basic building blocks of deep learning are activation functions and layers. Activation functions determine the output of computational graph nodes (i.e., neurons in neural networks). They can be linear or non-linear functions. Some popular activation functions include linear, sigmoid,tanh, ReLU (Rectified Linear Unit) and its variants. A layer in neural networks refers to neurons stacked in some forms to adapt the type of input data. Commonly-used layers include fully connected, convolutional & pooling, and recurrent layers. One can leverage those layers to build different popular neural networks architectures. For example, multilayer perceptron (MLP) networks are composed of fully connected layers, convolutional neural networks (CNN) are featured by varying groups of convolutional & pooling layers, and recurrent neural networks (RNN), and long short term memory (LSTM), are built upon recurrent layers Goodfellow u. a. (2016); Pang u. a. (2021). And more recent advanced architecture can combine many layers types to enhance the performance of the model.

Given a dataset $X = \{x_1, x_2, ..., x_N\}$ with $x_i \in \mathbb{R}^p$, let $Z \in \mathbb{R}^d (d < p)$ be a representation space, then deep anomaly detection aims at learning a feature representation mapping function $f_{RM}(.) : \mathbb{R}^p \rightarrow \mathbb{R}^d$ or an anomaly score learning function $f_{AS}(.) : \mathbb{R}^p \rightarrow \mathbb{R}$ where anomalies can be easily differentiated from the normal data instances in the space yielded by the f_{RM} or f_{AS} function, where both are a neural network. In the case of learning the feature mapping f_{RM}, an additional step is required to calculate the anomaly score of each data instance in the new representation space or an additional feature reconstruction $f_{Rec}(.) : \mathbb{R}^d \rightarrow \mathbb{R}^p$ in the case of Auto-Encoders Rumelhart u. a. (1986); Baldi (2011); Goodfellow u. a. (2016) as an example. While f_{AS} can directly infer the anomaly scores with raw data inputs, binary calssifier for example. An overview of the deep anomaly detection consists of three conceptual paradigms (See Fig 3.7).

In (a) Fig 3.7, The task of representation Learning using Deep learning is not directly related to anomaly detection, where this method is used as some independent feature extractor only Dara und Tumma (2018); Rumelhart u. a. (1986). This category of methods aims at leveraging deep learning to extract low-dimensional feature representations from high-dimensional and/or non-linearly separable data for downstream anomaly detection. The feature extraction and the anomaly scoring are fully disjointed and independent from each other. Thus, the deep learning components work purely as dimensionality reduction only Pang u. a. (2021). Compared to traditional dimension reduction like principal component analysis (PCA) Candès u. a. (2011); Schölkopf u. a. (1997), deep learning tech-

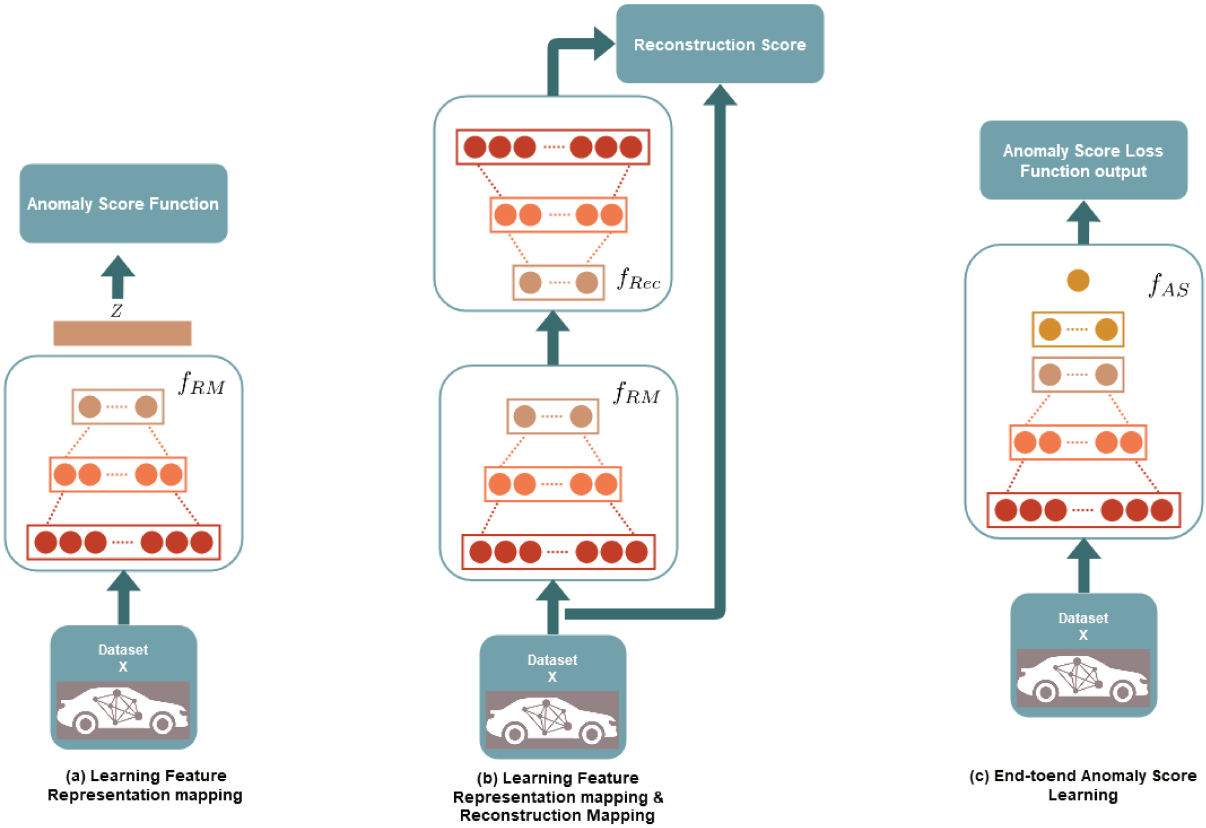


Figure 3.7: Conceptual Frameworks of Three Main Deep Anomaly Detection Approaches

niques have been demonstrating substantially better capability in extracting semantic features and non-linear feature relations Bengio u. a. (2013); Goodfellow u. a. (2016).

In the second category (b) in Fig 3.7, the objective is learning expressive representation of normality. This category of methods can be further divided into three subcategories based on how the representations are learned. The first category guides the learning with a *reconstruction loss* on the actual input data, as shown in the figure (b) 3.7. The second subcategory is based on *Adversarial Learning*; This approach aims to learn a latent feature space of a generative network. The latent space captures the normality underlying the given data. Some form of residual between the real instance and the generated instance is then defined as the anomaly score. We will study those approaches in details in the next chapter 4. The Third subcategory is *Predictability modelling* for sequence modelling, where the anomaly score is also a reconstruction loss. Still, the input is not given for the model (Forecasting). Predictability modelling-based methods learn feature representations by predicting the current data instances using the previous instances' representations within a temporal window as the context. In this section, data instances are referred to as individual elements in a sequence, e.g., network frames sequence in In-vehicle network communication. This technique is widely used for sequence representation learning and prediction. We will study those approaches in details in the next chapter 4 related to sequence modelling for In-vehicle CAN protocol. We note that each subcategory is taking a different approach in formulating its objective function.

In the third category (c) in Fig 3.7, neural network directly learns the anomaly scores. Novel loss functions are often required to drive the anomaly scoring network. Formally, this approach aims at learning an end-to-end anomaly score learning network in which the methods are dedicated to learning anomaly scores via neural networks in an end-to-end fashion binary classification (softmax likelihood models) or one-class classification. In Saxe u. a. (2019) provides an explanation for representation learning as the trade-off between finding a minimal compression Z of the input X while retaining the informativeness of Z for predicting the label Y. Put formally, supervised deep learning seeks to minimize the mutual information $I(X;Z)$ between the input X and the latent representation Z while maximizing the mutual information $I(Z;Y)$ between Z and the classification task Y.

3.3.2 Deep Anomly Detection Challenges

The above complex general problem discussed in Section 3.1.3 leads to several Deep Anomaly detection challenges. Some challenges, such as scalability data size, have been well addressed in recent years Pang u. a. (2021); Bulusu u. a. (2020); Chalapathy und Chawla (2019), while the following are remaining as challenges, to which deep anomaly detection can play some essential roles.

- **Anomaly Detection recall rate:** Since anomalies are infrequent and heterogeneous, it is difficult to identify all anomalies. Many normal instances are wrongly reported as anomalies (False-positive) while not reporting sophisticated abnormalities. Although plenty of anomaly detection methods has been introduced over the years, the current state of the art methods, especially unsupervised methods, still often produce high false positives on real-world datasets. Reducing false positives and enhancing detection recall rates is one of the most critical yet complex challenges, particularly for the future in-vehicle network anomalies.

- **Anomaly detection in high-dimensional and not-independent data:** Anomalies often exhibit evident abnormal characteristics in a low-dimensional space yet become hidden and unnoticeable in a high-dimensional space. High-dimensional anomaly detection has been a problem, especially in an environment with large amounts and modalities like intelligent vehicles. Performing anomaly detection in a reduced lower-dimensional space spanned by a small subset of original features or newly constructed features is a straightforward solution or feature selection-based method Keller u. a. (2012); Lazarevic und Kumar (2005); Liu u. a. (2012); Azmandian u. a. (2012); Pang u. a. (2018, 2017). However, identifying complex feature interactions and correlation in an end-to-end fashion may be essential in high-dimensional data and avoiding additional feature extraction process is important in the environment as embedded systems. Still, it remains a significant challenge for anomaly detection. Further, guaranteeing the new feature space preserved useful information for specific detection methods is critical to downstream accurate anomaly detection, but it is challenging due to the aforementioned unknowns and heterogeneities of anomalies.

- **Data Topology:** It is also challenging to detect anomalies from instances that may depend on each other, such as by temporal, spatial, graph-based, multi-modal, distributed and other interdependency relationships. Thus, Deep Learning architecture must fit the data topology needs.
- **Data-efficient learning while leveraging the small amount of anomalies:** Utilizing those labelled data to learn expressive representations of normality/abnormality is crucial for accurate anomaly detection. Weakly-supervised anomaly detection assumes we have some labels for anomaly classes and the class labels are partial (they do not represent the entire set of anomaly class), inexact (Noises and system fault instead of attacks). Two significant challenges are how to learn expressive normality/abnormality representations with a small amount of labelled anomaly data and how to learn detection models that are generalized to novel anomalies uncovered by the given labelled anomaly data.

Our main studies and contributions are motivated by the challenges mentioned above. In the following chapters 4,5,6 we take into consideration the different constraint related to the in-vehicle network system (Section 2.4.1).

3.4 Discussion

As our objective is an Anomaly detection based Intrusion detection system, we review the various efforts and research works relative to Anomaly Detection. Thus, including the emerging field of Deep Learning and its applications to Anomaly detection in many domains. Based on the SOA information, we will discuss the hypothesis and problematics that led us to our actual contributions. Our objective in the following chapters is to present our contributions in addressing the In-vehicle IDS challenges with Deep Anomaly Detection. We present our focus and work positioning compared to state of the art established in this chapter in Fig 3.8.

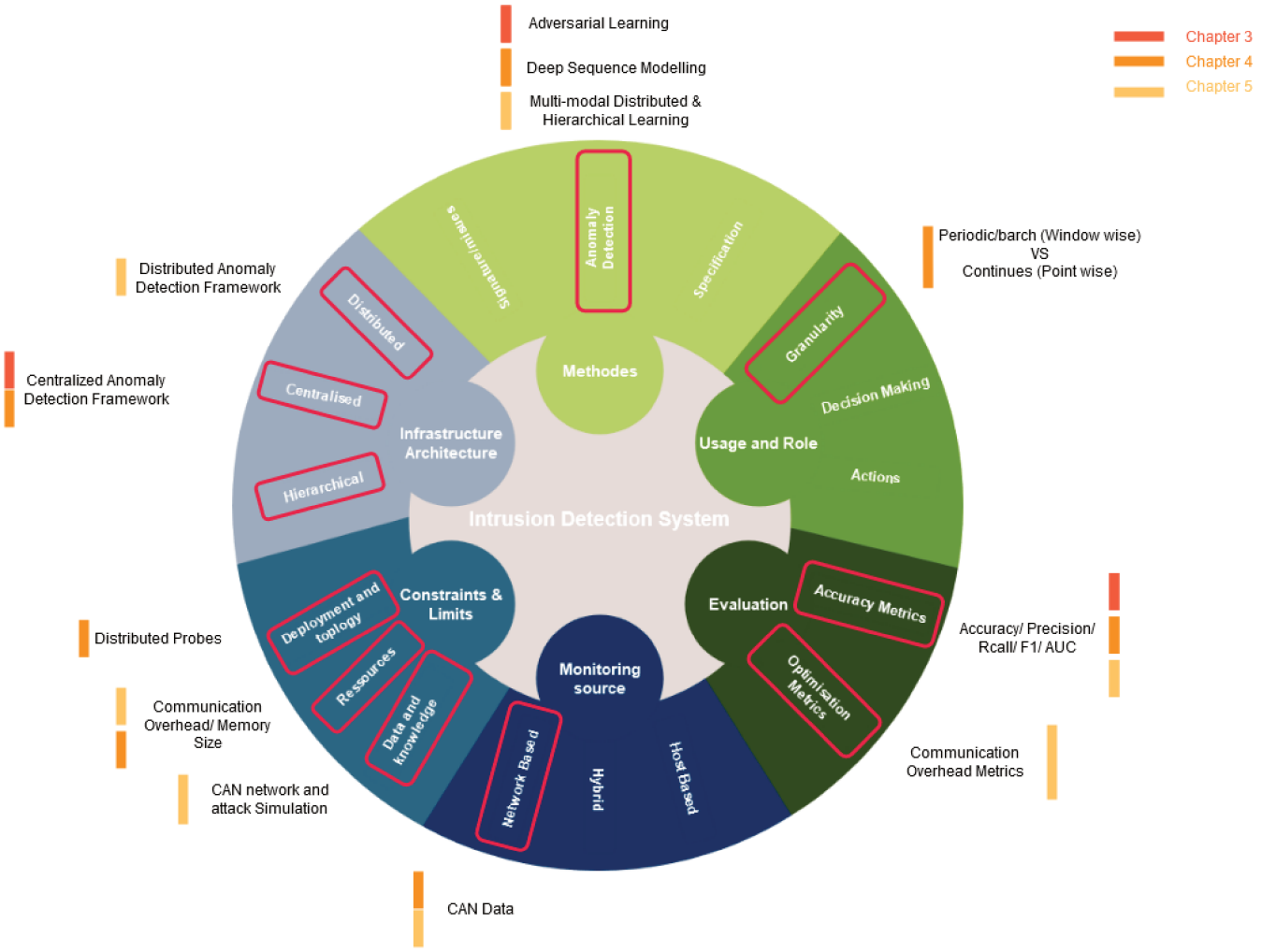


Figure 3.8: This Figure displays the different dimension of Intrusion Detection Taxonomy treated in this thesis with their main specific chapter of study and contribution.

Part II

Contributions: May we help your car to save your life?

Chapter 4

Encoding Adversarial Network for anomaly detection (AnoEAN)

In this chapter, we review the Generative Adversarial Networks (GANs) in detail by discussing the strength of the GANs compared to other generative models, how GANs works, and some of the significant problems with training, tuning and evaluating GANs. We also present some major GANs based architectures to show how a specific design helps solve various applications issues. All this is done to develop an Adversarial Learning approach to respond to the Anomaly detection-based IDS challenges (Introduced in the previous chapter in Section 3.3.2), more specifically, a lightweight memory-based approach. The emergence of the Generative Adversarial Networks (GANs) has recently brought new algorithms for anomaly detection. Most of them use the generator as a proxy for the reconstruction loss. We review the different GANs based architectures and processes used for Anomaly Detection. We will point out the major constraints of those approaches applied as a model for the Intrusion Detection system. We develop an alternative approach for anomaly detection, based on an Encoding Adversarial Network (AnoEAN), which maps the data to a latent space (decision space). The detection of anomalies is done directly by calculating a score. Our encoder is learned by adversarial learning, using two loss functions, which constrain the encoder to project regular data into a Gaussian distribution. The second is to project anomalous data outside this distribution. We conduct a series of experiments on several standard bases, network data and image data. This work has led to the publication of two conference papers:

- Elies Gherbi, Blaise Hanczar, Jean-Christophe Janodet, Witold Klaudel. An Encoding Adversarial Network for Anomaly Detection. 11th Asian Conference on Machine Learning (ACML 2019), Nov 2019, Nagoya, Japan. pp.1–16.
- Elies Gherbi, Blaise Hanczar, Jean-Christophe Janodet, Witold Klaudel. Construction d'espace latentpour la détection d'anomalies par apprentissage adversarial. Conférence sur l'Apprentissage automa-tique (CAP

2019), Jul 2019, Toulouse, France.

4.1 Introduction

In the previous chapter Section 3.3, we introduce Anomaly detection as a well-established topic in Machine Learning, with many applications in domains such as fraud detection, cybersecurity, video surveillance, and predictive maintenance (Chandola u. a., 2009; Kiran u. a., 2018; Hodge und Austin, 2004; Chalapathy und Chawla, 2019). Moreover, the recent advances in Artificial Intelligence bring insight into future applications, as autonomous transportation like self-driving cars. Many cyber-security threats can impact the usability of those systems (Chapter 1). The problem of anomaly detection highlights many risks related to those threats and can help the standard security systems to face new threats.

We present three main settings for anomaly detection. The first is the unsupervised case, where an algorithm has to discover the data's intrinsic properties to detect the anomalies without any label guidance(Campello u. a., 2015; Kiran u. a., 2018); The training set contains both normal and anomalous examples, but the labels are not available. The second is the supervised case, where an algorithm must usually face unbalanced datasets, with significant rates of normal data and few anomalous examples. The third is the one-class classification, where the algorithm has access only to a large set of normal data. Unlike the unsupervised case, there are no anomalous examples in the training set. In this chapter, we focus on both the supervised (unbalanced) and one-class anomaly detection problems. Notice that even though few anomalous examples may exist in the data, supervised learning algorithms are still challenging because the set of anomalies often does not form a homogeneous class. The boundary between normal and anomalous examples is blurred and continuously evolving (the system's normal behaviour changes depending on the context). This boundary problem is a challenge in the anomaly detection problem.

We can formulate the anomaly detection problem as follows. Let D be a data set containing a large number of normal examples (the normal states of the system) X_n, and a relatively small number of anomalous examples X_a. A model M must learn the distribution function p_X over the normal data during training. Then, given any test example x, it must determine whether x deviates from the learned distribution p_X by using an anomaly score function $a(x)$. The model M also needs to consider its number of parameters θ_M that represents the memory size of a Deep Learning model.

The balance of the instance numbers among classes in a dataset impacts Machine learning and Deep Learning-based classification performance. Indeed, in the cases of suffering from the data imbalance problem, the number of training samples belonging to a normal class is larger than anomalous classes. It's a common issue in Anomaly Detection, especially for cybersecurity. This data related problem impacts the machine learning algorithms and deteriorates the classifiers effectiveness Kozik und Choraś (2016). Typically, classifiers will achieve higher predictive

accuracy over the majority class but more insufficient predictive accuracy over the minority class. Utilizing unsuitable evaluation metrics for the classifier trained with the imbalanced data can lead to wrong conclusions about the classifier's effectiveness. As most machine learning algorithms do not operate very well with imbalanced datasets, the commonly observed scenario would be the classifier ignoring the minority class. This happens because the classifier is not sufficiently penalized for the misclassification of the data samples belonging to the minority class. Unbalanced data is well studied in our experiments. The goal is to develop a method that can benefit from the availability of a low rate of anomalies to enhance the model's capacity to detect different variant of anomalous examples and even unknown attacks.

Deep neural networks performs well in learning highly complex and large data representations.A large number of papers propose Deep Learning-based anomaly detection models (Goodfellow u. a., 2016; Hodge und Austin, 2004; Pimentel u. a., 2014; Bulusu u. a., 2020; Pang u. a., 2021; Chalapathy u. a., 2019; Chalapathy und Chawla, 2019; Kiran u. a., 2018). The recent advances in Deep Learning have made it possible to revise this problem and in many application domains like intrusion detection system Galinina u. a. (2018). In particular, Generative Adversarial Networks (GANs) (Goodfellow u. a., 2014), which were proved very efficient in many application fields (Creswell u. a., 2018), have also been adopted in recent works on anomaly detection (Schlegl u. a., 2017; Zenati u. a., 2018b,a; Akcay u. a., 2019; Golan und El-Yaniv, 2018; Sabokrou u. a., 2018; Deecke u. a., 2018; Mattia u. a., 2019; Zenati u. a., 2018a; Schlegl u. a., 2019).

In this chapter, we investigate several problems related to GAN-based anomaly detection methods. We propose a new method, called AnoEAN (Encoding Adversarial Network for Anomaly Detection). The principle of AnoEAN is to learn a function (Encoder) that projects the original dataset into a small dimension latent space so that the normal examples are projected in a restricted region of the latent space and the anomalies outside this region. This latent representation allows us to identify anomalies directly in the latent space by using a Mahalanobis distance on the distribution induced by the normal examples. We thus eliminate all the problems related to the reconstruction loss function. To do so, we develop a new approach that trains an encoder by adversarial learning. We assume that we have a large amount of normal data and a small number of anomalies, which is a common framework for anomaly detection. We finally conduct a series of experiments proving that AnoEAN performs better than conventional anomaly detection techniques, including those based on GANs, using both the MNIST base of handwritten digits (LeCun, 1998) and two standard network intrusion detection databases (KDD'99, NSL-KDD).

4.2 Generative Adversarial Network : Background and related works

Research in Deep neural networks has resulted in the design and development of different deep neural networks classes (Discriminative, Generative). The goal of the generative model is to learn the density function that describes the original sample data and then uses this estimated density to generate fake yet realistic-looking data similar

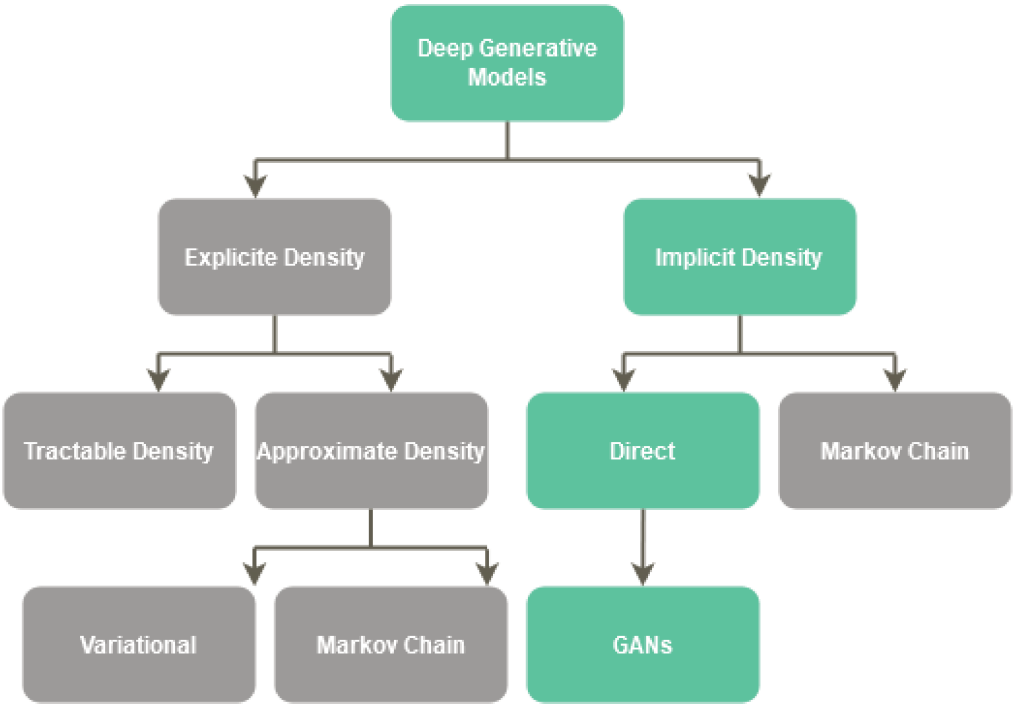


Figure 4.1: Schema distinguishing the two main learning process of generative models Goodfellow u. a. (2016); Goodfellow (2017); Goodfellow u. a. (2014)

to the original sample Goodfellow u. a. (2016). The Fig 4.1 distinguish two main learning process. Generative models that can learn via the maximum likelihood principle differ concerning how they represent or approximate the likelihood. On the left branch of this taxonomic tree, models construct an explicit density, maximizing the explicit likelihood. Among these explicit density models, the density may be computationally tractable, or it may be intractable, meaning that to maximize the likelihood, it is necessary to make either variational approximations or Monte Carlo approximations (or both). On the tree right branch, the model does not explicitly represent a probability distribution over the space where the data lies. Instead, the model provides some way of interacting less directly with this probability distribution. Typically the indirect means of interacting with the probability distribution is the ability to draw samples from it. Some of these implicit models that offer the ability to sample from the distribution are 1) MarkovChain; the model defines a way to stochastically transform an existing sample to obtain another sample from the same distribution. 2) Generative Adversarial Networks (GANS).

The increasing improvement in modern optimization and regulation techniques led to the emerging of Generative Adversarial Networks (GANs) Goodfellow u. a. (2014). GANs directly estimates a density function over a data distribution using alternative training techniques *Adversarial Learning*. GAN's core idea is to train two adversarial networks, a generator and a discriminator, in the form of a minimax game. The generator's goal is to generate realistic images that can fool the discriminator. In contrast, the discriminator tries to classify generated images (generated by the generator network) as fake and classify the real images from the original sample as real Goodfellow u. a. (2014). The adversarial Learning process aims for the generator to capture the distribution of true examples

enabling new data example generation. The discriminator is usually a binary classifier, discriminating generated examples from the true examples as accurately as possible.

4.2.1 GAN algorithm

Adversarial Learning Process

A Generative Adversarial Network (GAN) is composed of two neural networks (See Fig 4.2): a generator G, that transforms a vector z drawn from simple prior distribution (latent space) into an artificial data space (same dimension as a real data space), and a discriminator D whose role is to differentiate artificial data from real data. One drives such a system competitively: the generator must deceive the discriminator by implicitly learning to approximate the distribution of real data (Goodfellow u. a., 2014). More formal, GANs represent a mapping from noise space to data space as $G(z, \theta_g)$, where G is a differentiable function represented by a neural network with parameters θ_g. The Discriminator $D(x, \theta_d)$ is also defined with parameters θ_d and the output of $D(x)$ is a single scalar. $D(x)$ denotes the probability that x was from the data rather than the generator G. The discriminator trained to maximize the probability of giving the correct label to both training data and fake samples generated from the generator G is trained to minimize $\log(1 - D(G(z)))$ simultaneously.

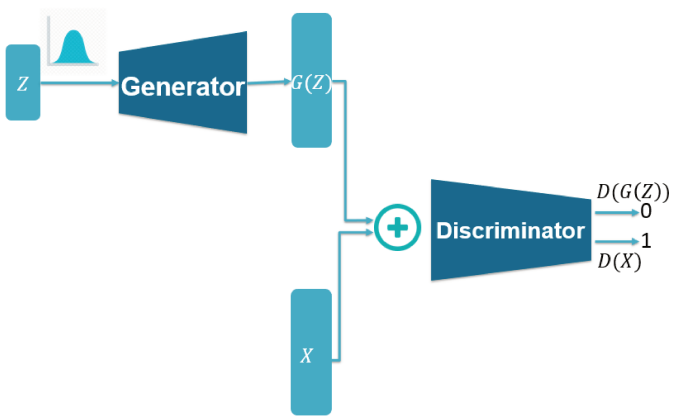


Figure 4.2: The Structure of GAN, Given a sampled noise from P_z (defined and known distribution), the generator objective is to convince the discriminator that its fake generated data is real. The discriminator take both real data and fake data. The role of the discriminator is to classify the real data as real (1) and generated data as fake (0).

If the input data is from the real data x, the discriminator learns to make $D(x)$ close to 1. If the input data is from the generated data $G(z)$, the discriminator strives to make $D(G(z))$ close to 0 while the generator G tries to make it close to 1. Since this is a zero-sum game between G and D, the optimization of GAN can be formulated as a min-max problem in the following objective function:

$$\min_G \max_D V(D, G) = \mathbf{E}_{x \sim p_{data}}[\log(D(x))] + \mathbf{E}_{z \sim p_z}[\log(1 - D(G(z)))] \tag{4.1}$$

It is possible that the Equation 4.1 cannot provide sufficient gradient for G to learn well in practice. Generally speaking, G is poor in early learning, and samples are clearly different from the training data. Therefore,D can reject the generated samples with high confidence. In this situation,$\log(1-D(G(z)))$ saturates. We can train G to maximize $log(D(G(z)))$ rather than minimize $log(1 - D(G(z)))$. Furthermore, there are other possible ways of approximating maximum likelihood within the GANs framework. A comparison of original zero-sum game, non-saturating game, and maximum likelihood game is shown in Fig 4.3 Nowozin u. a. (2016); Gonog und Zhou (2019).

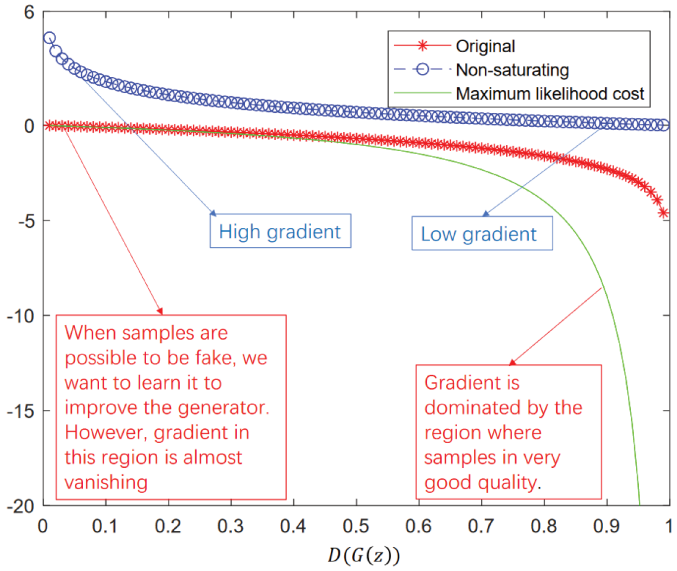


Figure 4.3: The three curves of “Original”, “Non-saturating”, and“Maximum likelihood cost” Gonog und Zhou (2019)

GANs Training process

Fig 4.2, consists of sampling a minibatch from both the training set and the generated samples and then running the discriminator on those inputs. We begin by sampling the latent vector z from the prior distribution over the latent variables (noise vector). It allows the generator to output a wide variety of different vectors Goodfellow u. a. (2014); Goodfellow (2017). We then apply the generator function to the vector z. The generator outputs a sample that is then applied to the discriminator. The discriminator outputs a value that is essentially a binary classification of real or fake. The error loss on the discriminator output is calculated using a cross-entropy cost function. This error is then backpropagated to both the generator and the discriminator networks. Training stops when the discriminator can no longer discriminate between generated data and training data. This point is known as the saddle point of the discriminator loss to describe the equilibrium, and in theory, should be the global minimum.

Training GANs in practice show some instabilities in reaching the saddle point. Several ways through which improvements to the GANs performance has been studied Salimans u. a. (2016). It has been shown that labels may improve subjective sample quality because the ability to learn a conditional model $p(y/x)$ often gives much better samples from all classes than learning $p(x)$ does. Also, batch normalization in the generator G can cause a

strong intra-batch correlation. Balancing the neural network architectures of the Discriminator and Generator may aid overall performance.

GANs Advantages

GANs were proposed to overcome the disadvantages of other generative algorithms. The basic idea behind adversarial learning is that the generator tries to create realistic examples to deceive the discriminator. The discriminator tries to distinguish fake examples from true examples. Both the generator and discriminator improve through adversarial learning. This adversarial process gives GANs notable advantages over other generative algorithms. The basic properties set GANs aside from some of the other generative models. We outline these properties as follows Goodfellow (2017):

- Implicitly modelling the high dimensional distribution of data.
- GANs have been proven to do well in high dimensional spaces Radford u. a. (2016) without relying on Markov chain Monte Carlo techniques.
- GANs excels in various challenging tasks, such as realistic image generation Brock u. a. (2018); Karras u. a. (2017), video frame generation Vondrick u. a. (2016), artistic style migration.
- The latent variable can be used to describes everything that is generated later by the generator network. This property is common with VAEs and Boltzmann machines.
- It can produce better samples than other models (the image is sharper and clearer).
- GANs can train any generative network. Most other frameworks require the generative network to have some specific functional form.

4.2.2 GANs variants and Anomaly detection application

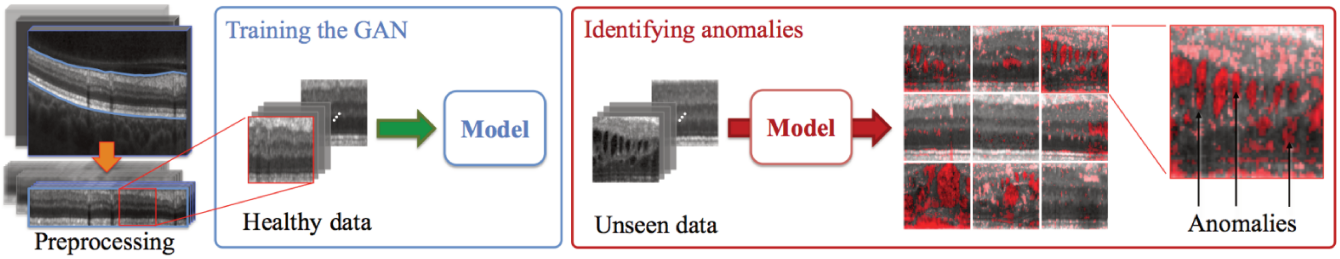


Figure 4.4: The GAN is trained on positive samples. At test time, after optimization process iteration, the latent vector that maps the test image to its latent representation is found z_{opt}. The reconstructed image $G(z_{opt})$ is used to localize the anomalous regions.

(a) original data (b) reconstructed (c) original data (d) reconstructed

Figure 4.5: (a) and (b) shows the non-identifiability issue: the generator performs well in reconstruction, but the encoder has difficulties to retrieve the correct pairs between the data space and latent space. (c) and (d) shows the mode collapse issue: the generator is more likely to project every latent point to digit (6), which proves that the generator lacks from diversity.

GANs have been successfully applied to model complex and high dimensional distribution of real-world data in many domains and applications Gonog und Zhou (2019). This GAN characteristic suggests they can be used successfully for anomaly detection, although their application has been only recently explored. Anomaly detection using GANs is the task of modelling normal behaviour using the adversarial training process and detecting the anomalies by measuring an anomaly score. Anomaly detection using GANs is an emerging research field. Schlegl u. a. (2017) propose AnoGAN, the first GAN based Anomaly Detection. AnoGAN learns a GAN from the normal data. Then, for a given test example x, the algorithm looks for a point z in the latent space such that $G(z) \approx x$; if it fails, then the example x is considered as an anomaly. In other words, the GAN-based approaches work upon the inversion of the generator: AnoGAN tries to compute $z = G^{-1}(x)$ under the assumption that G is invertible only for normal examples. Nevertheless, AnoGAN does not explicitly invert the generator: a reconstruction loss function is defined, typically $\mathcal{L}_r = \|x - G(z)\|$, and the techniques consist in finding the point z which minimizes $\mathcal{L}_r$ with respect to z. The process is very time-consuming because a gradient descent is performed for each test example. Notice that Creswell und Bharath (2019) improved this idea by inverting many images at once. The GANs also suffer from the mode collapse (Salimans u. a., 2016): The generator may have a limited diversity during the inference phase regardless of the input. In practice, it means that there exists a single point that the generator thinks is the most optimal point to generate, whatever the input. See Fig. 4.5 (c) & (d).In AnoGAN, the authors showed that GANs could be used for anomaly detection by introducing an optimisation process to find the optimal latent variable for a given input data (See Fig. 4.4). But these optimisation steps for every input lead to bad performance in inference time, and it is not suitable for application like Intrusion detection systems.

In order to get around these problems, other algorithms like EGBAD and ALAD (Zenati u. a., 2018a,b) learn Bidirectional GANs (BiGANs) (Donahue u. a., 2016; Dumoulin u. a., 2016; Li u. a., 2017) rather than GANs. Their architectures are composed of three networks (as illustrated in Fig. 4.6): a generator G and a discriminator D as standard GAN, and a third network called an encoder E, whose role is to project the data to the latent space with $E = G^{-1}$. The encoder is learned alongside the GAN. Moreover, in this setting, finding a point z such that $G(z) \approx x$ is immediate since we have $z = E(x)$. However, training an additional encoder network may encourage over-fitting, resulting in poor reconstruction (Creswell und Bharath, 2019). Moreover, in practice, the generator and the

encoder are rarely exact inverses of each other: they are inverse from a theoretical standpoint, after convergence. Furthermore, the BiGANs may suffer from non-identifiability issues, as shown by Li u. a. (2017): a single instance z can potentially correspond to many possible values of x (see Fig. 4.5 (a) & (b)). To deal with this problem, ALICE (Li u. a., 2017) uses conditional entropies to control the uncertainty of a pair (z, x). More precisely, the authors bound the conditional entropy by using the criterion of cycle-consistency (Zhu u. a., 2017).

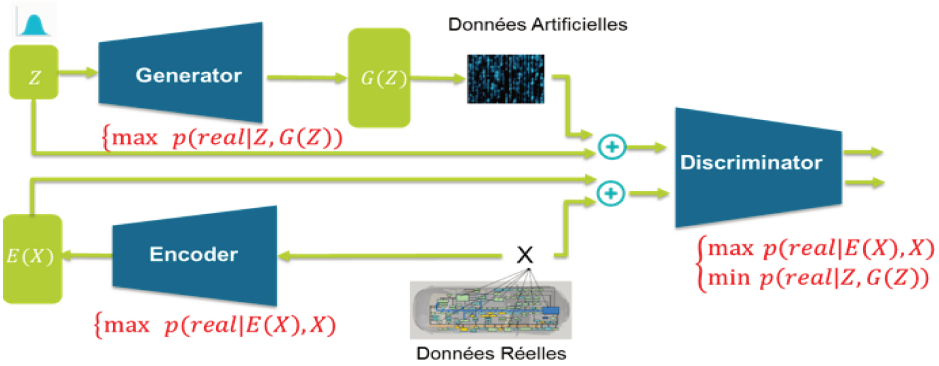


Figure 4.6: The structure of BIGAN

In Akcay u. a. (2019), the authors train a generator network on normal samples to learn their manifold (Latent representation Z). Simultaneously, an autoencoder is trained to efficiently encode the images in their latent representation (see Fig. 4.7). This work intends to enhance manifold learning using an Autoencoder to give more semantic proprieties to the latent space instead of prior gaussian noise. The generator network consists of three elements: an encoder G_E a decoder G_D (both assembling an autoencoder structure), and another encoder E. The architecture of the two encoders is the same. G_E takes in input an image x and outputs an encoded version z of it. Hence, z is the input of G_D that outputs$\hat{x}$, the reconstructed version of x. Finally, $\hat{x}$ is given as an input to the encoder E that produces $\hat{z}$. When the input data x is an anomaly, its reconstruction will be normal. Because the generator will always produce a non-anomalous image, the visual difference between the input x and the produced $\hat{x}$ will be high and will spatially highlight where the anomalies are located. The encoder E at the end of the generator structure helps, during the training phase, to learn to encode the images in order to have the best possible representation of x that could lead to its reconstruction $\hat{x}$. This approach implies the use of several objective functions, making it difficult to stabilise the general losses. Besides, using data space reconstruction (image) and latent space reconstruction doesn't match experimentally, meaning that a higher anomaly score, that's computed only in the latent space, can be associated with a generated sample with a low contextual loss value and thus very similar to the input Mattia u. a. (2019). The other point is that the proposed model uses four neural networks in both training and inference to enable anomaly detection. It costs more memory, which impacts their utility, especially in the embedded systems environment.

Most of the GAN-based approaches work upon the assumption that the generator is invertible. The inverse of the

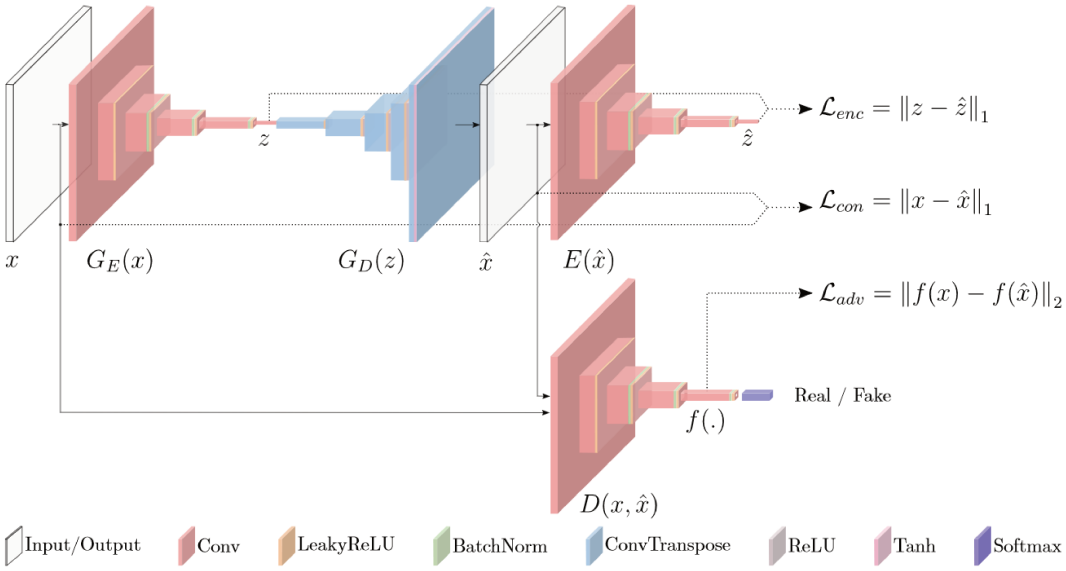


Figure 4.7: GANomaly architecture and loss functions from Akcay u. a. (2019).

generator is obtained either by driving another network that simulates the generator's inverse or via an optimization process. However, experimentally, it has been shown that in both cases, the approximated function is not the inverse of the generator everywhere (see Fig.4.5 (a) and (b)). Thus, applying the generator again on the antecedent sample generally fails to reconstruct x, even though the sample x was a normal example. In practice, this phenomenon dramatically increases the rate of false alarms.

4.3 Encoding adversarial network for Anomaly Detection

4.3.1 Formulation

We consider a problem in which we monitor the data $x \in \mathbb{R}^p$ describing the state of a system with p variables. Our goal is to trigger an alert when the data shows that the system is out of normal behavior. For this, we construct an anomaly score $a(x) : \mathbb{R}^p \rightarrow [0, +\infty[$ measuring the degree of anomaly of an example x, the normal example must have a score close to 0. To learn this function, we have a training set containing N examples corresponding to a normal behavior $\{x_{n_i} \mid i = 1..N\}$ and M examples corresponding to anomalies $\{x_{a_i} \mid i = 1..M\}$ with $M \ll N$. Notice that unlike normal data, the anomalies do not form a homogeneous class.

Our method, named AnoEAN for *Anormaly Detection by Encoding Adversarial Network*, consists in projecting the examples x_n and x_a into a small space, called decision space, in which the distribution of normal examples is Gaussian. We call *encoder*, the neural network that projects the examples from the original space into the decision space $E(x) : \mathbb{R}^p \rightarrow \mathbb{R}^d$ with $d \ll p$. Let p_z be a Gaussian distribution $\mathcal{N}(0, I)$ in the decision space. The purpose of the encoder is to project the normal examples in p_z and the anomalies outside p_z.

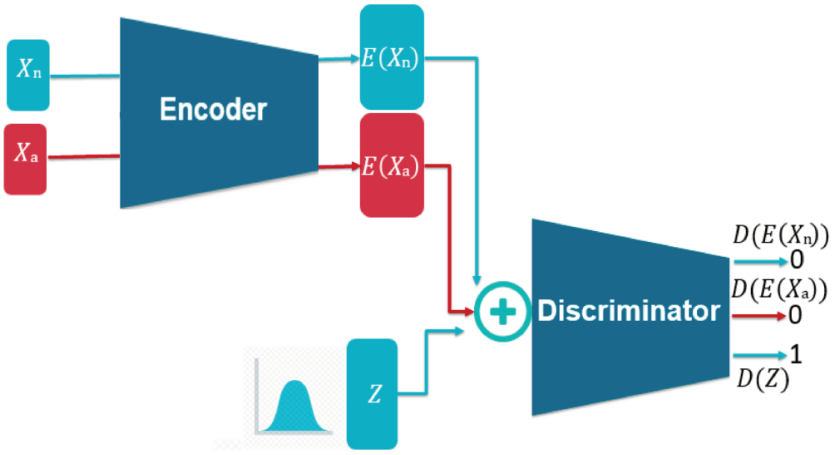


Figure 4.8: The training phase of AnoEAN: the Encoder and the Discriminator are adversarially learnt.

For this, in the same way as the GANs, we use a second network called *discriminator* $D(z) : \mathbb{R}^d \rightarrow \mathbb{R}$ which receives as input a vector of the decision space and predicts if this vector comes from p_z or from the encoder. $D(z)$ returns the probability that the vector z comes from p_z. The encoder must therefore both misleads the discriminator on the normal example projection $E(x_n)$ to make it believe that it comes from p_z and help the discriminator differentiate p_z from the projection of the anomalies $E(x_a)$. The architecture of our AnoEAN model is shown in Fig. 4.8. The discriminator and the encoder must respectively minimize the following L_D and L_E loss functions:

$$L_D = - \mathbf{E}_{z \sim p_z}[\log D(z)] - \mathbf{E}_{x_n \sim p_{x_n}}[\log(1 - D(E(x_n)))] - \mathbf{E}_{x_a \sim p_{x_a}}[\log(1 - D(E(x_a)))] \tag{4.2}$$

$$L_E = \mathbf{E}_{x_n \sim p_{x_n}}[\log(1 - D(E(x_n)))] + \mathbf{E}_{x_a \sim p_{x_a}}[\log(D(E(x_a)))] \tag{4.3}$$

In Eq. (4.2) and (4.3), the two first terms of loss function L_D, and the first term of loss function L_E are similar to the basic loss functions of a GAN. Instead of a noise vector in the input of the generator of the GAN, in AnoEAN a vector representing a normal example is in the input of the encoder. By maximizing $D(z)$ and minimizing $D(E(x_n))$, the decoder learns to differentiate vectors drawn from p_z and projections of normal examples $E(x_n)$. By maximizing $D(E(x_n))$ the encoder gets the distribution of the projections of normal examples $E(x_n)$ closer to p_z. If we do not add the terms $D(E(x_a))$ in L_D and L_E then there are no constraints on the projections of the anomalies, and the encoder tends to project both normal examples and anomalies on p_z. The consequence is that we can not differentiate normal examples from anomalies in the decision space. On the contrary, by minimizing $D(E(x_a))$, we get that (i) the decoder learns to differentiate vectors drawn from p_z and projections of normal examples $E(x_a)$, and

(ii) the encoder also minimizes the overlap between the distribution of projections of the anomalies $E(x_a)$ and p_z.

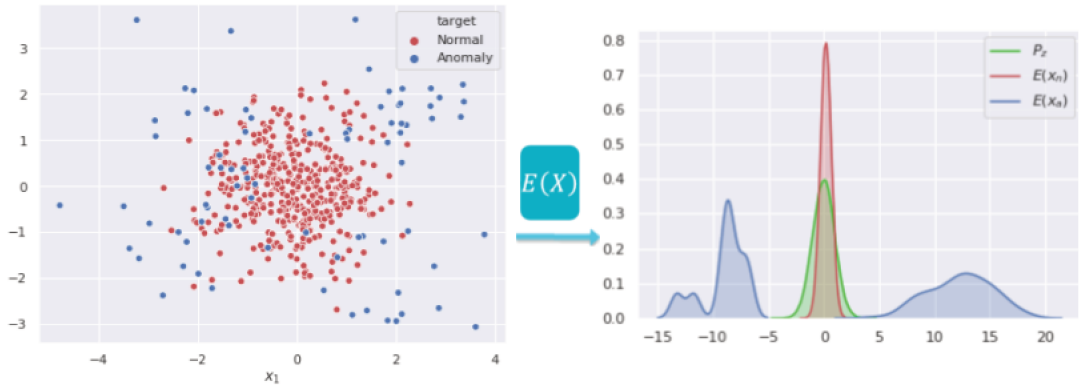


Figure 4.9: A sketch of the projection from the data space to the decision space that is performed by a well-trained Encoder.

Fig. 4.9 shows a toy dataset illustrating the objective of the encoder. The 2-dimension dataset (left panel) corresponds of 550 training examples; The normal examples are represented with 500 red points, which are drawn from a normal distribution; The remaining 50 blue points are drawn from multiple other Gaussian distributions and scattered around to simulate anomalous examples. The 1-dimension decision space is represented in the right panel, where the green, red and blue curves represent respectively the distribution of p_z, $E(x_n)$ and $E(x_a)$. The distribution of the $E(x_n)$ fits p_z and the overlap between the distribution of the $E(x_a)$ and p_z tends to 0. This clearly yields an efficient space in which the anomaly score can be computed.

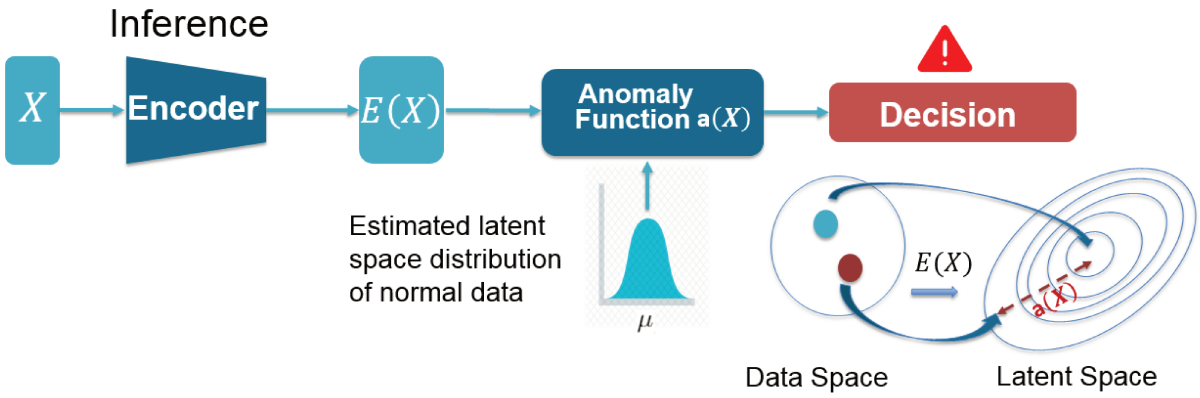


Figure 4.10: The inference (test) phase: only the Encoder is requested.

Once the model is trained, the prediction of the anomalies requires only the use of the encoder (Illustrated in Fig 4.10). The anomaly score for an example x is the distance between its projection in the decision space $E(x)$ and the distribution of the projection of the normal examples. This distribution is supposed to tend to $p_z = \mathcal{N}(0, I)$ during the learning of the model. We could therefore use the $E(x)$ standard deviation as an anomaly score. However,

because of the finite size of the training set, our experiments showed that the projection distribution of normal examples could diverge slightly from p_z. Therefore, at the end of the learning, we represent this distribution by a Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ whose parameters are assessed with a validation base. The anomaly score is finally the Mahalanobis distance between $E(x)$ and μ:

$$a(x) = \sqrt{(E(x) - \mu)^T \Sigma^{-1} (E(x) - \mu)} \tag{4.4}$$

4.3.2 Theoretical analysis

In this section, we develop the loss functions L_D and L_E (see Eq. (4.2) and (4.3)) in order to identify the optimal discriminator and encoder. We begin with the discriminator. The formulas of L_D can be rewritten by introducing the integrals instead of the expectations:

$$\begin{aligned} L_D &= \int_z -p_z(z) \log D(z) dz - \int_x p_{x_n}(z) \log(1 - D(E(x))) dx - \int_x p_{x_a}(x) \log(1 - D(E(x))) dx \\ &= \int_z -p_z(z) \log D(z) - p_{z_n}(z) \log(1 - D(z)) - p_{z_a}(z) \log(1 - D(z)) dz \end{aligned}$$

where p_{z_n} (resp. p_{z_a}) is the distribution of the projection of normal (resp. anomalous) examples into the decision space. We can express the optimal discriminator D given the encoder E. Let $f(D(z))$ be the function in the integral above such that $L_D = \int_z f(D(z)) dz$. To find the discriminator that minimized L_D , we set the derivative of $f(D(z))$ to 0 :

$$\begin{aligned} \frac{\partial f(D(z))}{\partial D(z)} = -\frac{p_z(z)}{D(z)} + \frac{p_n(z)}{1 - D(z)} + \frac{p_a(z)}{1 - D(z)} = 0 \\ \Rightarrow D(z) = \frac{p_z(z)}{p_z(z) + p_{z_n}(z) + p_{z_a}(z)} \end{aligned} \tag{4.5}$$

if $p_z(z) + p_{z_n}(z) + p_{z_a}(z) \neq 0$, we can compute the second derivative to check that this extremum is a minimum :

$$\frac{\partial f(D(z))}{\partial D(z)} = \frac{p_z(z)}{D(z)^2} + \frac{p_n(z)}{(1 - D(z))^2} + \frac{p_a(z)}{(1 - D(z))^2} > 0 \tag{4.6}$$

Hence the optimal discriminator is $D^*(z) = \frac{p_z(z)}{p_z(z) + p_{z_n}(z) + p_{z_a}(z)}$. It corresponds to the Bayes classifier predicting if z comes from the distribution p_z or from the encoder.

Let us see what the encoder does when the optimal discriminator is plugged into loss function L_E. We have:

$$
\begin{aligned}
L_E &= \int_z p_{z_n}(z)\log(1-D^*(z)) + p_{z_a}(z)\log(D^*(z))dz \\
&= \int_z p_{z_n}(z)\log\left(\frac{p_{z_n}(z)+p_{z_a}(z)}{p_z(z)+p_{z_n}(z)+p_{z_a}(z)}\right) + p_{z_a}(z)\log\left(\frac{p_z(z)}{p_z(z)+p_{z_n}(z)+p_{z_a}(z)}\right)dz \\
&= \int_z p_{z_n}(z)\log\left(\frac{p_{z_n}(z)+p_{z_a}(z)}{p_{z_n}(z)}\frac{p_{z_n}(z)}{p_z(z)+p_{z_n}(z)+p_{z_a}(z)}\right) + p_{z_a}(z)\log\left(\frac{p_z(z)}{p_{z_a}(z)}\frac{p_{z_a}(z)}{p_z(z)+p_{z_n}(z)+p_{z_a}(z)}\right)dz \\
&= -\int_z p_{z_n}(z)\log\left(\frac{p_{z_n}(z)}{\frac{p_{z_n}(z)+p_{z_a}(z)}{2}}\right)dz - \log 2 + \int_z p_{z_n}(z)\log\left(\frac{p_{z_n}(z)}{\frac{p_z(z)+p_{z_n}(z)+p_{z_a}(z)}{3}}\right)dz - \log 3 \\
&\quad - \int_z p_{z_a}(z)\log\left(\frac{p_{z_a}(z)}{p_z(z)}\right)dz + \int_z p_{z_a}(z)\log\left(\frac{p_{z_a}(z)}{\frac{p_z(z)+p_{z_a}(z)+p_{z_a}(z)}{3}}\right)dz - \log 3 \\
&= -KL\left(p_{z_n}\middle|\middle|\frac{p_{z_n}+p_{z_a}}{2}\right) + KL\left(p_{z_n}\middle|\middle|\frac{p_z+p_{z_n}+p_{z_a}}{3}\right) - KL\left(p_{z_a}||p_z\right) + KL\left(p_{z_a}\middle|\middle|\frac{p_z+p_{z_n}+p_{z_a}}{3}\right) \\
&\quad - 2\log 3 - \log 2
\end{aligned}
\tag{4.7}
$$

Clearly, the optimal encoder minimizes an expression composed by 4 Kullback-Leibler divergences. The minimization of the both terms $KL\left(p_{z_n}\middle|\middle|\frac{p_z+p_{z_n}+p_{z_a}}{3}\right)$ and $KL\left(p_{z_a}\middle|\middle|\frac{p_z+p_{z_n}+p_{z_a}}{3}\right)$ implies that the encoder tends to the solution where the three distribution are equal $p_z = p_{z_n} = p_{z_a}$. By maximizing $KL\left(p_{z_a}||p_z\right)$, the encoder maximizes the divergence between p_{z_a} and p_z i.e. it tries to project the anomalies outside p_z. To maximize $KL\left(p_{z_n}\middle|\middle|\frac{p_{z_n}+p_{z_a}}{2}\right)$, the encoder has to maximize the divergence between p_{z_n} and p_{z_a}, thus it aims at separating the projection of the normal examples from the projection of the anomalies. By optimizing all of these four divergences together the encoder tries to project the normal examples into p_z and to project the anomalies outside p_z and p_{x_n}.

4.3.3 Learning algorithm

Algorithm 1 AnoEAN. Adam hyper-parameters ($\alpha = 0.0002, \beta = 0.5$), Learning rate 10^{-3}

Require: batch size m, discriminator step n_D, $m = m_a + m_n$
Initialize the discriminator parameters w_D, the generator parameters θ_G.
for number of training iteration **do**
  sample batches of $\{x_n^{(i)}\}^{m_n} \sim P_{x_n}$, $\{x_a^{(i)}\}^{m_a} \sim P_{x_a}$, $\{z^{(i)}\}^{m} \sim \mathcal{N}(\mu, \sigma^2))$.
  for $k = 0, ..., n_D$ **do**
    $\delta_w \leftarrow \nabla_w[-\frac{1}{m}\sum_{i=1}^{m}\log D_w(z^{(i)}) - \frac{1}{m_n}\sum_{i=1}^{m_n}\log(1 - D_w(E_\theta(x_n^{(i)}))) - \frac{1}{m_a}\sum_{i=1}^{m_a}\log(1 - D_w(E_\theta(x_a^{(i)})))]$
    $w_D \leftarrow Adam(\delta_w, \alpha, \beta)$
  end for
  $\delta_\theta \leftarrow \nabla_\theta[-\frac{1}{m_n}\sum_{i=1}^{m_n}\log D_w(E_\theta(x_n^{(i)})) - \frac{1}{m_a}\sum_{i=1}^{m_a}\log(1 - D_w(E_\theta(x_a^{(i)})))]$
  $\theta_G \leftarrow Adam(\delta_\theta, \alpha, \beta)$
end for

In Algo. 1, we show the training procedure of our model. We take random examples x_n and anomalies x_a from the training set and project them into the decision space with the encoder (*i.e.*, we calculate each $E(x_a)$ and $E(x_n)$).

We add random vectors from the p_z distribution to these projected examples, and get the *batch* that is presented at the input of the discriminator. The discriminator is modified by a gradient descent to minimize L_D(See Eq 4.2); the encoder is frozen during this step. Then, it is the encoder's turn to be modified in order to minimize L_E(See Eq 4.3), the discriminator being frozen during this step. This procedure is iterated until convergence. Notice that to seed up the gradient descent, we modify the loss function of the encoder : $L_E = -\mathbf{E}_{x_n \sim p_{x_n}}[\log(D(E(x_n)))] - \mathbf{E}_{x_a \sim p_{x_a}}[\log(1 - D(E(x_a)))]$. The minimization of this function is theoretically equivalent of the previous one, but it has larger gradient for bad encoders.

4.4 Experiments

This section evaluates our method's performance (AnoEAN) on the anomaly detection task and compares it against other GANs methods and Classical OC-SVM. We describe our experimental protocol, the algorithms used to compare our model, the datasets, and our technique's implementation details.

The purpose of the evaluation is to show the performance of EAN on the anomaly detection task, especially in a different setting, where the anomalies are rare. Also, the ability to detect unknown attacks while dealing with a normal class with many modalities to demonstrate our model's efficiency compared to state of the art.

4.4.1 Experimental setup

In our experiments, we used three datasets: MNIST, KDD99, and NSL-KDD.

MNIST (LeCun, 1998) is a database of handwritten digits commonly used for image classification problems. Despite that MNIST does not initially contain normal classes and anomalies, it is often used in anomaly detection to analyze algorithms' behaviour and visualize their predictions. In our experiments, we used this dataset in two ways: 1) **1 against all**: we consider that a certain number represents the normal class and the remaining nine numbers represent the anomaly class. For the OCSVM, AnoGAN, ALAD, EGBAD methods, the training set consists of 5000 normal data; the test set includes 1700 examples, of which 80% of normal data and 20% of anomalies randomly selected among the other nine classes. For AnoEAN and SVM, we additionally inject 10% anomalies into the learning set; these are chosen among four classes out of nine so that certain figures (anomalies) do not appear during the learning.

2) **n against m**: it is based on the same principle as the "1 against all", the only difference being that the normal class is composed of several digits. We group n classes of digits into one normal class and treat the remaining m digits as abnormal examples. We use the same sample apportionment in the learning and test sets as in "1 against all". The objective is to analyze the algorithms' behaviour in the case where the normal class is heterogeneous and can be separated into several subclasses.

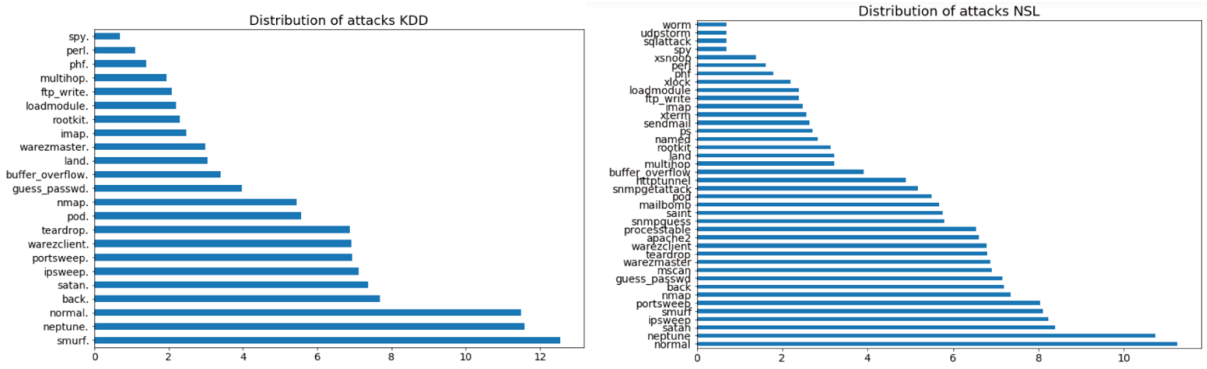


Figure 4.11: normal and attack examples distribution on both KDD and NSL-KDD datasets.

The other datasets KDD99 and NSL-KDD are commonly used to evaluate anomaly detection algorithms' performance, the objective being to detect intrusions into monitored networks. We use the same sample apportionment in the learning and test sets as before.

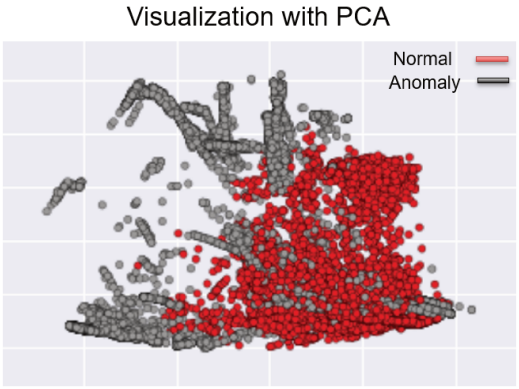


Figure 4.12: Visualization using PCA on NSL-KDD.We can see that the normal behavior has a strong pattern where most of the points are closer to each other, we see also that a lot of anomaly points overlaps with normal point pattern.

4.4.2 base-line

We compare our method with two kernel methods and three GAN-based methods.

OCSVM. The One-Class Support Vector Machine (OCSVM) is a classical kernel-based technique for novelty detection. It is usually used with the RBF or linear kernel function (Schölkopf u. a., 1999). In our experiments, the RBF kernel gives the best results. The OCSVM learns a decision boundary around normal examples, containing most of the learning samples. Samples residing outside the borders are considered abnormal. We implemented two tests, one where we feed OCSVM with normal data (fixing ν = 0.0001), the other where we inject 10% anomalous example in train data and a soft margin of 10% (that is, $\nu = 0.1$). The latter did not bring any improvement to the

results.

SVM. Although the SVMs often produce efficient solutions for balanced data sets, they are sensitive to unbalanced data sets (Veropoulos u. a., 1999; Wu und Chang, 2003; Akbani u. a., 2004). The main reason is that the objective function assigns the same cost C for positive and negative classification errors in terms of penalty (Veropoulos u. a., 1999). We used a cost-sensitive learning solution by adding a weight for each class that penalizes errors (giving a higher weight to the least frequent class corresponding to the anomalies class).

AnoGAN was the first published anomaly detection method based on GAN (Schlegl u. a., 2017). The model learns a generator able to project random points from a low-dimensional multivariate Gaussian distribution (latent space) to the distribution of the training data set. The model adversarially learns a discriminator that must separate the generated data from the real data. After the training stage, we sample a latent space variable using a generator, followed by gradient descent on this latent variable, to estimate the inverse projection for each element of the test set. The anomaly criterion combines the reconstruction loss and the feature match loss (distance computed with the discriminator's last hidden layer). For each test element, if the optimization cannot find a latent variable that minimizes the criterion, we obtain a high score.

EGBAD takes advantage of the BiGAN/ALI network structure. EGBAD has an encoder that projects the data into in the latent space (Donahue u. a., 2016; Dumoulin u. a., 2016). The adversarial training process is driven through a discriminator that takes as an input the pair (data, latent variable) and must determine which pair constitutes a real pair consisting of a sample of real data and its coding $(x, E(x))$, or a false data sample and the corresponding latent space input of the generator $(G(z), z)$. For a given test input x, EGBAD (Zenati u. a., 2018a) uses the encoder to infer the latent variable $z = E(x)$ that will be used as input for the generator to reconstruct the test input $G(E(x))$. The anomaly criterion is the same as that of Schlegl u. a. (2017).

ALAD is a bidirectional GAN, based on the ALICE architecture (Li u. a., 2017). It directly infers the reconstruction of data in the test phase using an encoder and the generator $G(E(x))$. In ALAD (Zenati u. a., 2018b), the authors regularize the conditional distribution by adding another discriminant $D_{xx}(x, G(z))$ to approximate a conditional entropy constraint. The aim is to enhance EGBAD by explicitly forcing the encoder and generator during the training so that both are inverse. The authors also apply the spectral normalization (Miyato u. a., 2018) to regularize the training. The anomaly score is the L_1 reconstruction error in the discriminant space G_{xx} between the actual data and the sample generated using the discriminant hidden layer before the logit layer.

OCSVM, ALAD, AnoGAN and EGBAD are one-class methods: Only normal data is used in the learning phase. In AnoEAN and SVM, we use few additional anomalies, leading to unbalanced classes.

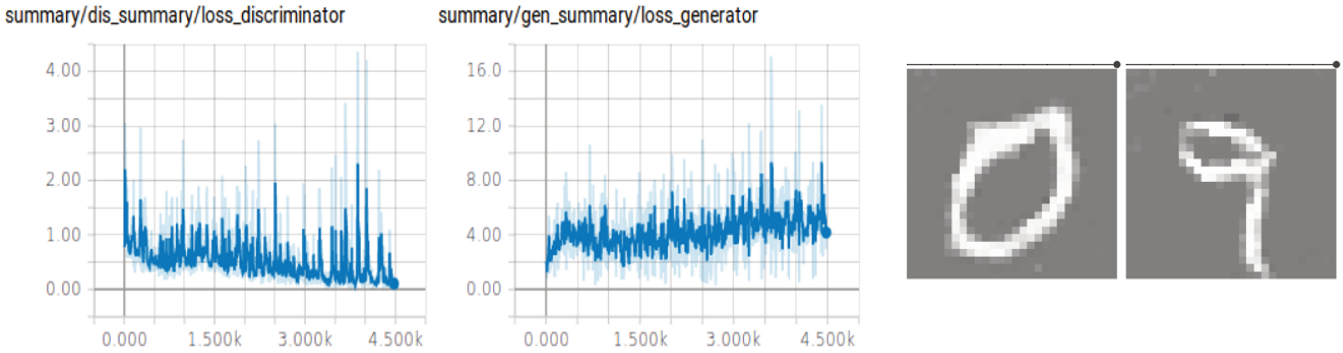


Figure 4.13: Loss on vanilla GAN without Spectral normalization.

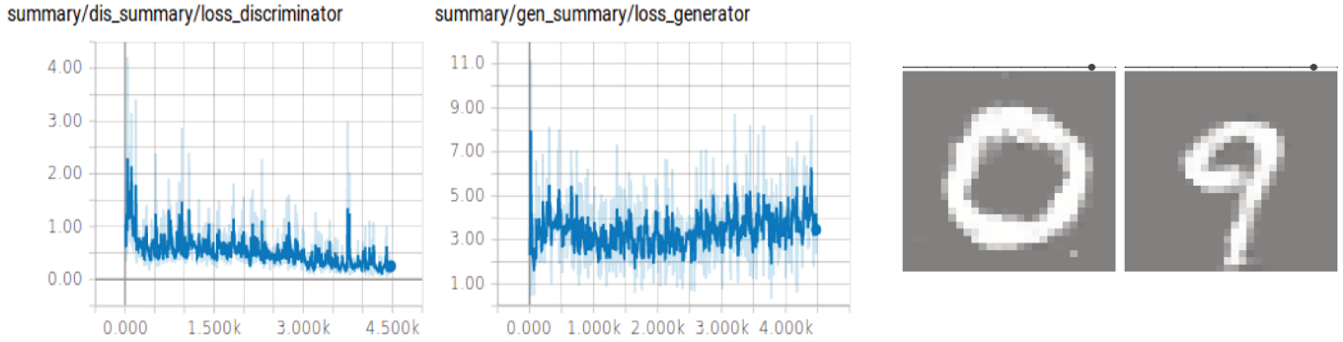
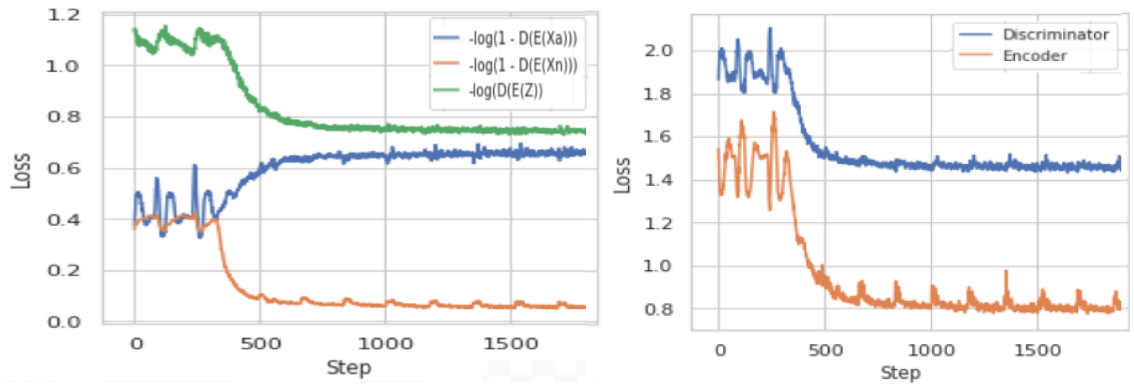


Figure 4.14: Loss on vanilla GAN using Spectral Normalization.

4.4.3 Training and configuration parameters

AnoEAN has two neural networks, the encoder and the discriminator. In the case of MNIST, the encoder is a convolutional network $(4*2*32, 4*2*64, 4*2*128, d)$(d is the dimension of latent space) and the discriminator is a multi-layer network $(32, 16, 1)$. For the NSL-KDD and KDD99 bases, the encoder and the discriminator are multi-layer networks $(128, 64, 32, d)$ and $(32, 16, 1)$ respectively. The size of the *batch* is 200. The number of *epochs* is fixed by *early stopping* and the gradient descent is performed with Adam (Learning rate starts at $lr = 10^{-3}$). We use spectral normalization Miyato u. a. (2018) to regularize each layer of the discriminator to control the Lipschitz constant of the layer, which can stabilize the training of GANs. Fig. 4.14 shows the impact of spectral normalization on GAN training and generation quality compared to Fig. 4.13 representing the vanilla GAN without spectral normalization. We notice that the loss value is lower and fluctuate less with the spectral normalization, showing more stability of the network. Besides, the image generated are have a better quality.

In Fig. 4.15(a), we show the curves of the loss function L_D. The green and blue curves show that the discriminator is confused through the learning steps: it cannot differentiate between the distributions over z and $E(x_n)$. Therefore, the encoder is getting better with respect to the approximation of P_z. At the same time, we see that the orange loss curve keeps decreasing, which means that the distribution of anomalous examples $E(x_a)$ diverges



(a) (b)

Figure 4.15: (a) shows the loss curves on the discriminator. (b) shows the overall loss functions of the encoder and the discriminator.

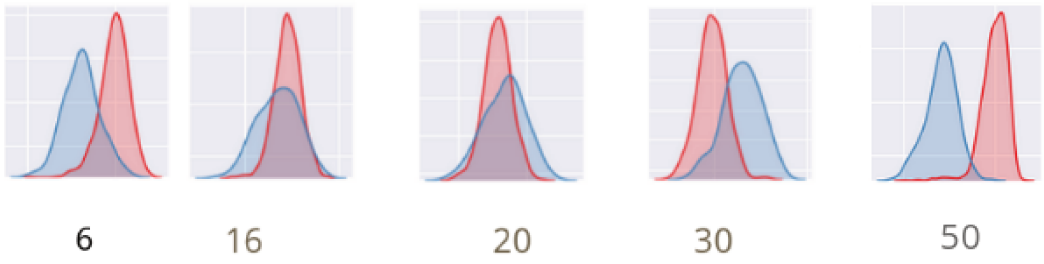


Figure 4.16: Visualisation of the encoder mapping test samples (blue for normal examples and red for anomalous examples) into the latent space.

from P_z. In Fig. 4.15(b), we can see that both the loss functions L_E and L_D are decreasing with the iterations until convergence. In Fig .4.16, we see the evolution of the encoder through different epochs. We can notice that the encoder is getting better at differentiating normal examples from anomalous example, meaning that the encoder is learning the mapping of normal example features into a Gaussian distribution. Thus, any example that doesn't match normal samples will not be near the gaussian centre.

4.4.4 Results

In Fig. 4.17, we present our results on MNIST (1 against all): each digit is successively considered as the normal class, the others being seen as anomalies. AnoEAN outperforms all other methods in 5 out of 10 cases; its performances are comparable to the best methods on the remaining 5 digits. Notice that all the methods have poor performances on Classes 2, 3, 4 and 5 because these classes have similar visual characteristics as those of the anomalies. We also observe that reconstruction-based approaches (ALAD and EGBAD) are competitive: They successfully rebuild the numbers, which allows them to have an important anomaly score on items that are not visually similar to the normal class.

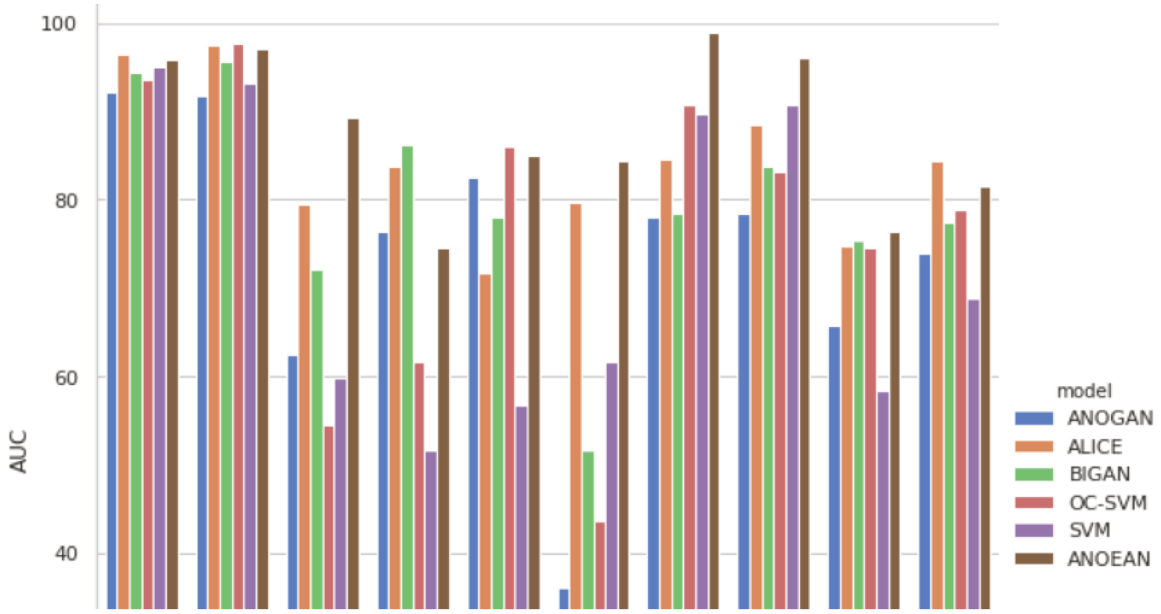


Figure 4.17: Model performance according to the AUC metric on each digit in the case of the MNIST problem (1 against all)

In Fig. 4.18, we show the results on MNIST (n against m). We are interested in the problem of identifying anomalies in a dataset where the normal class is heterogeneous (composed of several digits). In this configuration, AnoEAN dramatically outperforms the state of the art, even if its performance deteriorates with the increasing heterogeneity of the normal class. Notice that the performances of reconstruction-based approaches (ALAD and EGBAD) collapse. Indeed, for them to work, the generator and the encoder must necessarily be inverse from one another, what is particularly difficult to guaranty when the normal class is heterogeneous. On the other hand, AnoEAN succeeds in encoding complex distributions because it does not need to calculate the inverse of the encoder.

AUC	F1	ROC	accuracy	Model
81.3 ± 0.92	80.1±1.4	83.9±0.89	86.2±0.78	AnoGAN
95.1 ± 0.74	89.6±1.5	97.9±0.31	95.9±0.51	EGBAD
95.2 ± 0.77	90.2±0.81	98.0±0.39	96.1±0.28	ALAD
81.1 ± 4.52	78.5±4.98	91.1±2.24	88.3±7.48	OCSVM
97.9±0.26	95.2±0.63	98.8±0.28	98.3±0.25	**AnoEAN**
94.8 ± 0.5	93.9±0.8	95.6±1.49	97.4±0.24	SVM

Table 4.1: Experiments results on NSL-KDD dataset

We present the results on NSL-KDD and KDD99 datasets in Tables 4.1 and 4.2. We obtain these results by running the train and test with ten different seeds, for the initialization of the weights and shuffling before splitting data into train and test. The values represent the mean and standard deviation confidence intervals, to show a calibrated performance of the results. These tables show several measures of performances: the area under the precision-recall curve (AUC), the F1-measure (F1), the ROC curve, and the accuracy. Since the number of

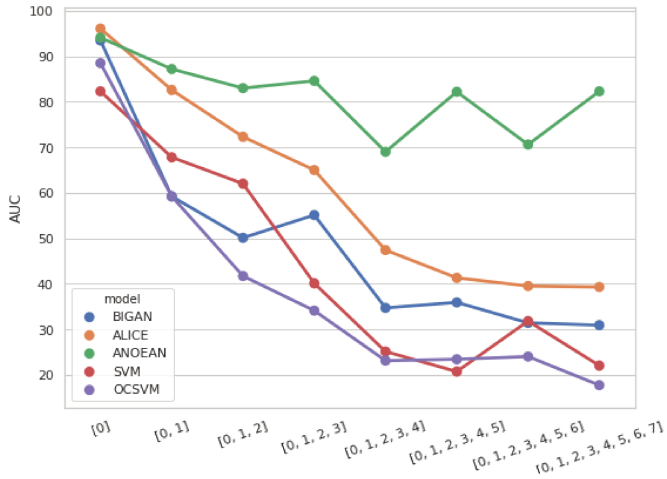


Figure 4.18: Visualization of the performances on different methods with several degrees of complexity in the normal class (varying the heterogeneity of the normal class with several numbers).

AUC	F1	ROC	accuracy	Model
82.7±2.72	85.9±1.97	85.8±0.79	88.1±1.71	AnoGAN
95.1±3.67	96.4±3.02	98.9±0.88	98.5±1.39	EGBAD
95.6±1.51	96.8±0.72	99.1±0.37	98.7±0.28	ALAD
86.6±3.86	84.4±5.36	95±2.02	92.5±3.21	OCSVM
99.2±0.08	97.3±0.32	99.7±0.09	98.8±0.14	**AnoEAN**
98.7±0	98.5±0	98.6±0	99.3±0	SVM

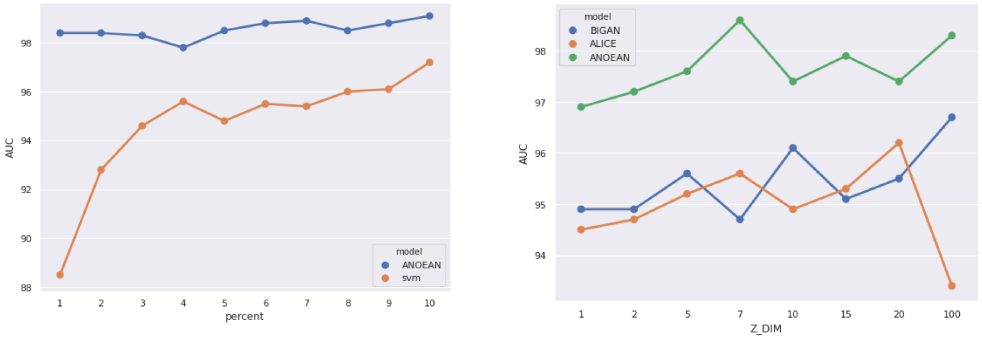
Table 4.2: Experiments results on KDD99 dataset

anomalies is much lower than the normal examples, AUC represents our primary measure of interest. Notice that ANOEAN has a better AUC than state of the art and also a better score on the other metrics on NSL-KDD. On the KDD dataset, ANOEAN is competitive with SVM method on F1 score and accuracy. Notice that KDD dataset contains more than 400 000 single connection vectors and even with 10% of anomalies, it remains a large amount of data; This is why the SVM has good results in that case. In Fig. 4.19(a), we reduce the amount of anomalies in the training set, and we verify that ANOEAN keeps on average the same performances, whereas those of the SVM decreases when the rate of anomalies decrease.

At last, in the Fig. 4.19(b), we check the impact of the latent space Z dimension on the AUC. Clearly, AnoEAN performs better than the other GANs based approach (EGBAD/ALAD) whichever the dimension of Z.

4.5 Conclusion

In this chapter, we presented a brief introduction to GANs methods and their application to anomaly detection, assess the GANs approach for anomaly detection based on reconstruction loss, their strength in learning the distribution in the context of One-Class classification using the generative approach. Our objective was to investigate



(a) (b)

Figure 4.19: (a) shows the impact of reducing the percentage of anomalies in AnoEAN and SVM on NSL-KDD. (b) shows the impact of the latent space vector dimension Z on the AUC metric, concerning the Adversarial Network Based models (BIGAN/ALICE/AnoEAN).

anomaly detection with the generative ability to learn about normal examples' distribution while using a few anomalous examples. We also keep in mind that the method is more designed for low computation systems like embedded systems like vehicles. The network size that needs to be used to infer the anomaly score needs to be relatively light. This is translated with a proposal of a new anomaly detection method that uses a projection in a latent space of small dimension in which normal examples and anomalies are separated. For this, we train an encoder by *adversarial learning*. The cost function enforces the encoder learning to projects the normal data into a Gaussian distribution and the anomalies in the tail of this distribution. For the prediction, our method only needs the encoder weights. It is crucial to have a small model for applications in embedded systems where the memory is costly. We have shown that our model could discover non-encountered anomalies during the learning stage. Besides, our technique performs better than the state of the art when faced with a heterogeneous normal class. Finally, compared to state of the art, our technique triggers few false alarms. Our method's limitation is that it needs an amount of anomalous example to restrain the projection of normal examples. In future works, we will adapt our method to the case where no anomalies are available in the training set. The idea is to complete the architecture with a new generator of artificial anomalies that should train against the encoder and the discriminator in an adversarial way to achieve this goal.

Chapter 5

Empirical Time Series Evaluation Of In-Vehicle Intrusion Detection System

In this chapter, we study the application of anomaly detection for in-vehicle Intrusion Detection System. To this end, we focus on CAN data as the primary data representing the In-vehicle's network behaviour. We simulate and analyse automotive CAN data to understand its nature and inner structure (functionally and statistically). We will present in this chapter a brief analysis of CAN data as Time series and discuss the impact of the various attacks on data. We review the related work on anomaly detection for CAN IDS, followed by our contribution. We introduce a multivariate time series representation for asynchronous CAN data, enhancing the temporal modelling of deep learning architectures for anomaly detection. We study different deep learning tasks (supervised/One-class) and compare several architectures to design an in-vehicle intrusion detection system that fits in-vehicle computational constraints. Our system is time window wise: any given time frame is labelled either anomalous or normal. We conduct experiments with many types of attacks on an in-vehicle CAN using the public SynCAn Dataset. We show that our system yields good results and allows us to detect many kinds of attacks. This work has led to the publication of a conference paper:

- Elies Gherbi, Blaise Hanczar, Jean-Christophe Janodet, Witold Klaudel. Deep Learning for In-Vehicle Intrusion Detection System. International Conference on Neural Information Processing (ICONIP 2020), Nov 2020, Thailand. pp.50–58.

5.1 Introduction

In recent years, a lot of effort has been made to build more intelligent vehicles and enable them to offer different services, ranging from automation and driver assistance to infotainment applications. To achieve this, microcontrollers

ECUs (Electronic Control Units) are deployed over the vehicle. They communicate with each other on the automotive bus system using various bus field busses (see Section.1.3.1), the main one being the Controller Area Network (CAN). Future applications like autonomous transportation require various technologies that allow the vehicles to interact with other vehicles (VANETs), pedestrians and road infrastructure. These controllers make the vehicles more connected with the external world, allowing new functionalities and dramatically increasing the security risk as exposed in the first chapter 2.3.

In this work, we focus on the CAN bus, which is *de facto* standard for in-vehicle communication. In-vehicle networks technologies must ensure a set of requirements, some of which are time-critical and safety-related. CAN protocol uses broadcast communication techniques. Each node in the network can send and receive a packet to/from the bus. CAN protocol increases the network elasticity and guarantees the latency time, and meets the real-time systems requirements Avatefipour und Malik (2017); It was designed to be lightweight, robust, and fast to meet the different in-vehicle systems and safety requirement. However, the CAN bus contains several vulnerabilities. It does not include the different security criteria in its design. It lacks security facilities like message authentication that prevents an unauthorised node from joining the communication and broadcast malicious messages to other nodes. It also lacks encryption because it would make overhead for real-time communication. The protocol's weaknesses are as many possibilities left to hackers to attack, as shown in the cyber-security literature Nilsson u. a. (2008); Avatefipour und Malik (2017). Several attacks scenarios have been demonstrated on vehicles. *E.g.*, Koscher u. a. (2010a) has performed four different tests on the control window lift, warning light, airbag control system and central gateway. Hoppe u. a. (2011) gives an experimental analysis of vehicle attack surface and wireless access; They inject an attack when the CD is played by the victim vehicle. Due to this, in-vehicle security has been paid more and more attention, and extensive research has been done to develop In-Vehicle intrusion detection systems, most of them using rule-based and statistical approaches.

We focus on anomaly detection based intrusion detection using the advances in deep-learning architectures to handle the CAN data structure. To do so, we define three levels for the in-vehicle IDS framework: 1) CAN data level, 2) sequence modelling level and 3) detection level. The CAN data level selects the information that the intrusion detection system is monitoring and how it is structured and organised to fit the sequence modelling. The sequence modelling level consists of the deep-learning models that can learn complex representations and hidden characteristics from the first level's time series data. The detection level uses this representation to detect anomalies by establishing a trust score for the actual events. The detection phase depends on the aim of the IDS and also on the type of available data. If we have times series labelled with known attacks, then we are in a supervised learning setting, and the goal of the model is to discriminate the attack patterns from the normal behaviour patterns. In the other case, the type of attacks is unknown, and no labels are available. In the One-class learning setting, the IDS has to learn the normal behaviour, and the goal is to detect any behaviour that deviates from the normal patterns.

This research aims to propose a general in-vehicle IDS using deep learning; we propose a representation for

CAN data and experiments several deep learning architectures for sequence modelling. We propose a practical and feasible anomaly detection IDS that meets the needs and constraints of in-vehicles systems. In the following section, we survey the related works about CAN data and intrusion detection systems using deep learning techniques. We present our main contribution in Section 3 and perform a large series of experiments showing our approach's relevance in Section 4, before concluding in Section 5.

5.2 Background and Related Works

5.2.1 CAN data

Figure 5.1: CAN message frame, We highlight the fields used to create the dataset. The ID defines the content, and Payload is the Encoded Signals. Each message is timestamped whenever the transmission is captured.

In Section 2.1.3 we introduce a functional characteristic of CAN. In this section, we analyze the CAN as input for machine learning algorithms. Any CAN message composed of several fields: an ID and a 64-bit payload. The ID is unique and defines the content (set of signals encoded in a range of several bits) and the priority of the message Farsi u. a. (1999). A timestamp is added whenever the message is captured. From those characteristics, many representation and feature can be derived as time interval, sequence, time-frequency. Notice that the payload signals are encoded, so it is hard to obtain the signals values without the constructor specifications.

CAN Intrusion Detection Systems generally intervene at two levels:

1) flow-based approach: In flow-based approaches, a flow is a group of packets sharing common properties during a specific period. Those properties can be defined as the meta-data with other statistical characteristics (*e.g.*, the number and the frequency of the messages). The model then learns the pattern of the CAN flow. The regularity of the in-vehicle ECUs communication makes it easy, in theory, to define a model that can spot any deviation from the original flow.

2) Payload based approach: In payload-based approaches, the payload represents the information carried by the packet. This information is exchanged between the ECUs, and their interpretation reflects the behaviour of the vehicle. In payload-based inspection, we learn a model that determines whether the different values are sound (for example, the engine speed and the vehicle speed).

Notice that some attacks may highly impact the communication flow, like flooding attack, and will be more easily detected at the flow level. On the other hand, other types of attacks are not visible in the communication flow and are seen only at the payload level. For example, when the attacker gains control on one of the ECUs, he can simulate

the same emission flow but contains wrong information.

5.2.2 CAN Intrusion detection System : related works

In-vehicle IDS may be either knowledge-based or anomaly-based (see Section 2.4). In knowledge-based approaches, the IDS use information about the attack and define a signature so that if a new event matches this signature, then an alarm is raised. These approaches are very efficient to deal with known attacks. As for anomaly-based approaches, the method starts by defining a model (*profile*) that specifies the system actual normal behaviour. Any behaviour that does not conform to the normal profile is considered an anomaly. The anomaly-based IDS have many advantages: there is no need to maintain a database of signatures, and they can detect unknown attacks since, at least from a theoretical standpoint, each attack compromises the normal behaviour of a system.

Dupont u. a. (2019) has provided an exhaustive survey about the actual CAN IDS and derived a comprehensive taxonomy from CAN data perspectives. The authors have proposed a well-fitting CAN IDS categorization by separating 1) the frame level (by message or sequential data), 2) the data source, and 3) the detection phase. The frame-level refers to the number of messages needed to build a model and detect an attack. A single message-based CAN IDS learns whether a frame is normal by using the information contained in this frame only. Kang und Kang (2016) uses deep belief networks to learn about normal packets' statistical properties and abnormal packets to discriminate between them. In Martinelli u. a. (2017), the payload of the message is used to classify a message as normal or abnormal. Other CAN IDS use information that can be retrieved from consecutive messages due to the regularity of ECUs communication; The time interval and ID's meta-data can be used as the frequency of IDs. In Young u. a. (2019), the authors propose a simple IDS based on time intervals, assuming that they are regular for each CAN ID. In Moore u. a. (2017), they also use time interval and frequency analysis, and they have shown that false-positive could increase due to some fluctuation caused by bus access negotiation.

With the advances of deep learning for time series, many deep learning architectures have been used to solve sequential modelling problems, and anomaly detection based CAN IDS is one of them. In Song u. a. (2020), the authors propose a deep convolutional network classifier IDS, a supervised approach designed to learn about traffic patterns and detect different malicious behaviours. They reduce the unnecessary complexity in the architecture of inception-resnet. They have tested their model on different types of attacks using bit-level CAN dataset where each input consists of 29 sequential CAN IDS. In Hanselmann u. a. (2020), the authors tackle a large dataset with an extensive type of attacks (SynCAN dataset). They propose a deep learning architecture that handles CAN data structure; It is composed of independent recurrent neural networks, one for each ID in the input. The goal of those separated LSTM is to learn about the state of each ID. The whole state of the network is represented by a joint vector of the outputs of all separated LSTM. The second part of the architecture takes the joint latent vector as an input for an autoencoder (fully connected network) that enables One-class learning; the autoencoder's task is to

reconstruct the signals for each possible input message based on the joint vector. The attack detection occurs by observing the deviation between the signal value of a message at the current time step with its reconstruction.

In Seo u. a. (2018), the author proposes a combination of two networks using only the IDs of CAN data: A normal discriminator (Binary classifier) to detect known attacks and a GAN-based IDS to detect the unknown attacks. Besides, they transform the input into a sparse image representation of CAN payload with one hot-encoding. Both networks' architecture is composed of a convolutional network, and the anomaly detection process has two steps. The first is to use the classifier to detect if the actual image (representation of successive CAN IDs) is abnormal, and if it is not, the CAN image representation is received by the discriminator learnt with the GAN to detect if the CAN image is a fake (Anomaly). Pawelec u. a. (2019) proposes an IDS by analyzing bit-level CAN message, using LSTM to predict the next message based on the history size of 10 messages; If the distance between the predicted message and the actual message is bigger than a threshold, then the message is an anomaly.

We note that, in the literature, many dimensions can be considered to design the CAN IDS. The used data highly impacts the type of detectable attacks and the building of the model that can learn about a broad range of situations to ensure that the model encompasses the exhaustive space of normality and decreases the false positive rate. There is also another dimension, which is the In-Vehicle context, where the memory and computation power is limited, so the practical feasibility of a given CAN IDS needs to be evaluated in front of the constraints of the in-vehicle context.

5.2.3 Contribution focus and positioning

In summary, the literature of CAN bus IDS strategy and methods are proposed based on how the attack manifest into the CAN bus network with manipulation on CAN frequency and CAN packet payload. In section 2.3 of the second chapter, we presented the recent studies concerning the different attack surfaces within the CAN network and its vulnerabilities Lee u. a. (2017); Taylor u. a. (2015); Lee u. a. (2015). Briefly, the CAN bus traffic is targetted in two ways, attack on CAN packet frequency and attack on CAN packet payload. The first one affects the flow of the traffic network, and the second affects the content and the semantic of the message. Indeed, the time interval between the CAN packet IDs is fixed and periodic (see Fig 5.2). Thus, the attacks that are involved in the CAN packet frequency is made by inserting an extra packet or erasing a legitimate packet from the CAN bus traffic by hijacking the timing of the transmission of the packets. On the other hand, attacks on the payload are made by manipulating or spoofing CAN IDs content. This kind of attack aims at changing the values within a packet content while maintaining the content valid, meaning that it would not produce any effects on the vehicle flow communication.

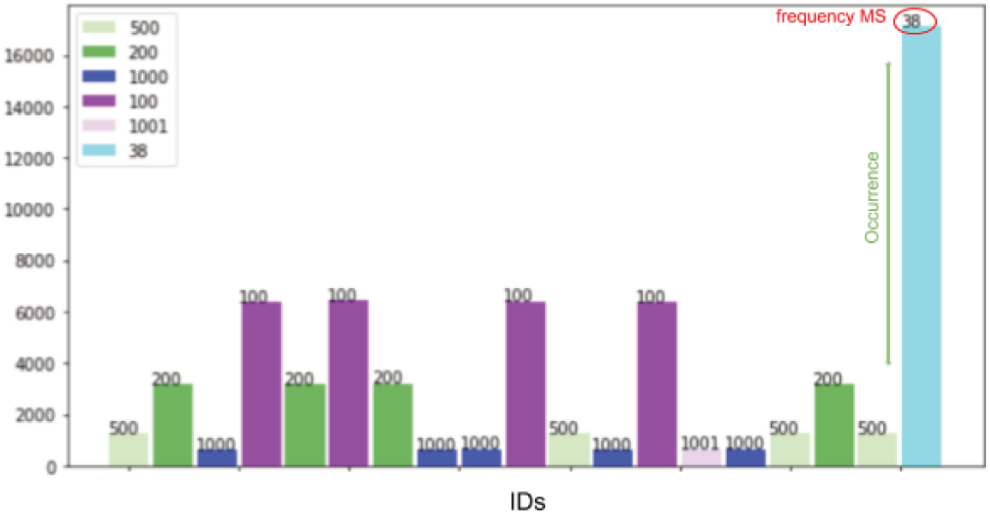


Figure 5.2: Histogram representing the number of occurrences of each ID in a given time-frame, and their frequencies.

In Fig. 5.3 We described a wide range of intrusion detection in the CAN bus system in three main levels that encompasses the primary focus of various research shaping the different in-vehicle IDS frameworks. Our work inclines towards an anomaly-based approach using deep learning because it is more suitable to learn from examples and intelligently adapt to the CAN environment's nature regardless of the protocol changes and dealing with novel attacks. Below we present the three different levels and our positioning.

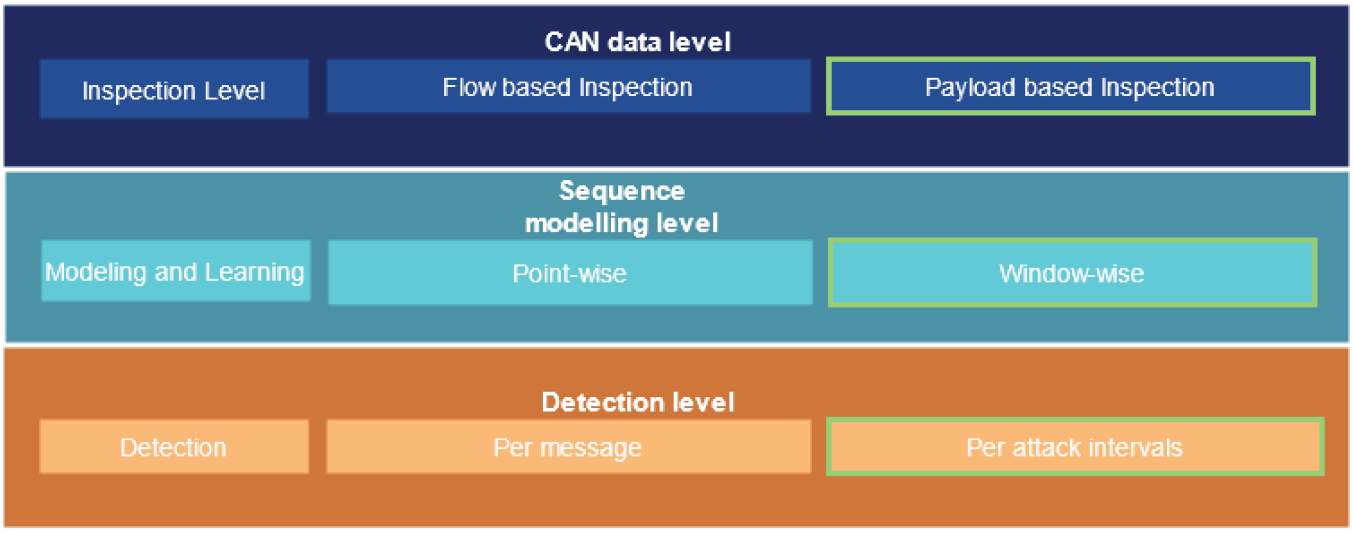


Figure 5.3: We show The different levels of an in-vehicle Intrusion Detection system. The highlighted squares define our choice and focus at each level on proposing a practical in-vehicle IDS for CAN data.

CAN data level

Concerning the **CAN data level** most techniques illustrated above in terms of inspection level are flow-based and use mainly the frequency as the main feature. This approach of detecting attacks is straightforward and legitimate due to the predictability of regular CAN bus traffic, which broadcasts packets at a fixed time interval. This approach can only detect attacks that alternate this fixed time interval, and also, technical issues in the network can occur, which will lead to false positives. The other methods that consider the payload feature inspection uses a specific feature (like speed engine pressure and engine speed, steering wheel, etc ...). In reality, most of those packets are encoded, and the message specification to obtain the values and features from the packet message is kept confidential by the constructors, and the available public datasets are encoded.

Most methods in the discussed studies focus on improving the core model by using Deep learning but neglect the examination of pre-processing parts. The input data need to consider the two inspection level so the core model can learn about the flow and the semantic content of the in-vehicle network. Also, The IDS overall performance is greatly influenced by the size of the data, primarily for CAN, as it broadcasts a large number of packets per second.

In **CAN data level** we propose a new representation that can handle the issues mentioned by grouping both content and frequency information. We will discuss the data construction in more detail in the section below 5.3.1.

Sequence modelling level

In this work, we focus on Deep learning for sequence modelling of time series. Thus our input is a sequence of different messages represented by their IDs. We define two modelling scenarios (*point-wise* and *window-wise*). The point-wise scenarios is when the model output score related to a single message or ID. The Window-wise is when the model's output is related to a sequence of messages IDs. the window-wise approach is more about giving the state of the network given a time-frame, and point-wise is about labelling each packet in the network if it normal or malicious. Obtaining an output for each data point is impractical, especially from a real-time CAN which generates a considerable volume of data in milliseconds.

We use both One-class and classification modelling to deal with both know and unknown attacks. We analyze in section 5.3.2 different deep learning architecture in terms of accuracy metrics and memory constraints. The results are discussed in the experimentation section 5.4.

Detection level

For the **detection level**, the window-wise modelling enables detecting attacks intervals. To detect the source and the main concerned ID or messages, it more of a forensic task. The proposed detection is about defining if the actual time frame is legitimate behaviour of the overall system or malicious behaviour.

5.3 Sequence modelling with Deep Learning for In-vehicle Intrusion Detection System

We propose a new IDS for the in-vehicle network described through 3 levels: 1) the CAN data level resumed in a feature matrix, 2) the time sequence modelling level, and 3) the anomaly detection level. The complete system is described in Fig. 5.4 and detailed below. Notice that only the online blocks functions are embedded in the vehicle and perform model inference and detection. The training and update of the models are performed offline and outside of the vehicle.

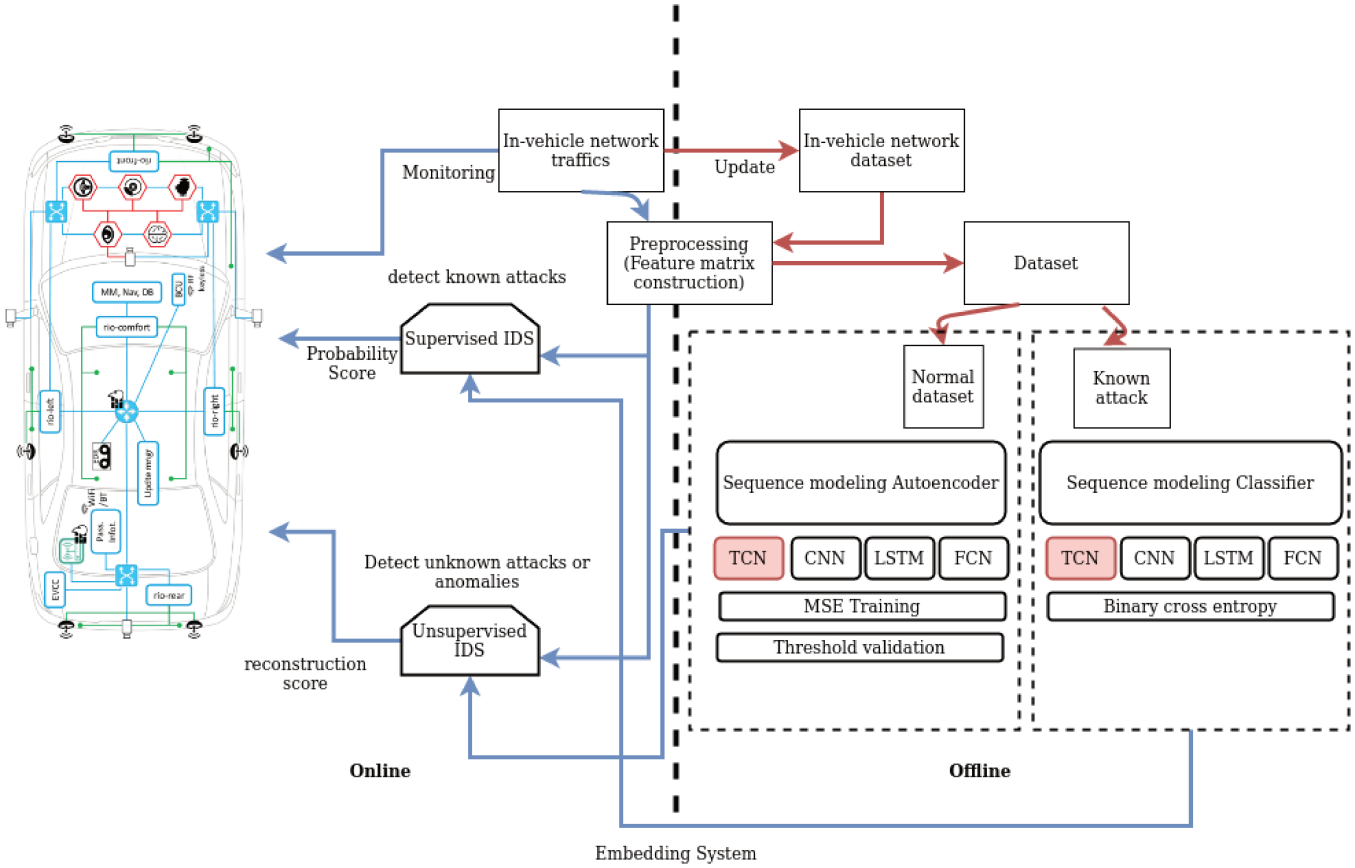


Figure 5.4: In-vehicle IDS workflow: The blue lines refer to online functions and the red lines for offline functions. The online blocks functions are about model inference and detction beside data transformation. The offline block contain more heavy functions as big data analysis, training and update the models.

5.3.1 Occurrence Matrix construction

The first step is to transcript the flow of CAN messages, sent separately and asynchronously at irregular moments, into a Multirate Multivariate times series (MMTS) which contains both payload and flow information. The CAN data is composed of different messages broadcasted from different ECUs. The stream S of messages is represented by a sequence of N messages $S = \{V_1, ..., V_N\}$. We denote $V_i = \{id_i, P_i, t_i\}$ the i-th message captured by the IDS.

$id_i \in \{1, ..., m\}$ is the ID of the ECU that sent the message, m is the number of ECU, P_i is the payload contained in the message and $t_i \in [0, t_N]$ is the time of reception. The payload P_i contains a vector of size n_{id} containing the signal sent from ECU id. The number of variables n_{id} extracted from each payload depends on the ECU. Although deep learning models can be trained with this type of sequences, this representation is clearly not optimal for neural networks learning. This representation's two main problems are that the same ECU's payload variables are not split around the sequence, and the time interval between two messages is not constant.

We change the messages flow representation S into a multivariate time series representation T. The messages flow's time range is discretized into K time stamps of constant time Δ . The time series $T = \{t_1, ..., t_K\}$ is a series of time points, each time point represents the set of messages received during the corresponding timestamp $t_k = \{V_t | t \in [(k-1)\Delta, k\Delta]\}$. This time series is then represented by a matrix $M_{\sum_{id=1}^{m}(n_{id}+1)\times K}$ where each column represents a time point, and the rows represent the variables contained in the payloads of all ECU. We could have received several messages from the same ECU for each time point and have different values for the same variables; In these cases, we keep the last received values since we want to decide on the most recently available information. We also add to this matrix a row for each ECU indicating the number of messages received from this ECU. This representation regroups both payload and flow features, enabling the model to detect both attacks on payload and flow in the in-vehicle communication system.

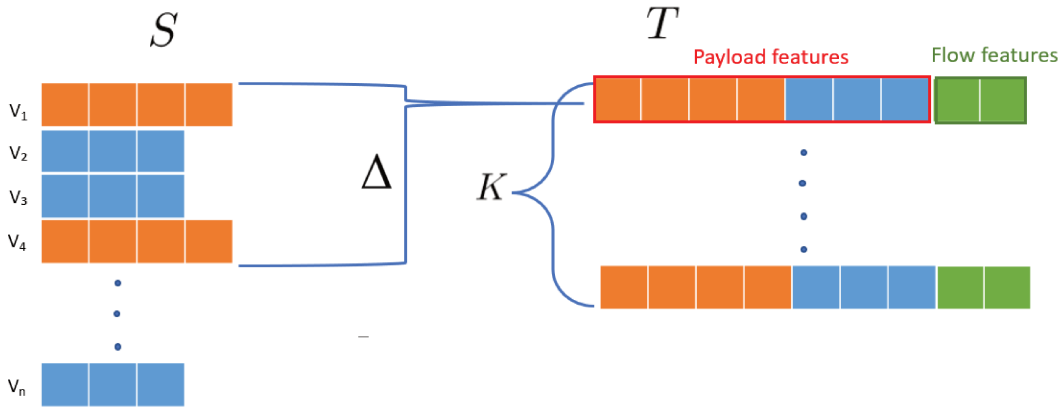


Figure 5.5: schema showing the transformation of the messages flow representation S into a multivariate time series representation T. The messages flow's time range is discretized into K time stamps of constant time Δ.

5.3.2 Deep Sequence modeling architectures

Many DNNs architectures are used for time series modelling tasks Ismail Fawaz u. a. (2019); Berman u. a. (2019); Han u. a. (2021). This work reviews four types of sequence modelling architectures: Fully-connected network (FCN), Recurrent neural network (LSTM), 1D Convolutional neural network (CNN) and Temporal convolution network (TCN). We compare their ability to learn the hierarchical representation vector of the CAN matrix to perform anomaly detection. This vector is either given by the autoencoders' bottleneck in the One-class learning task or the

final architectures layer in the supervised task.

Fully-connected network (FCN)

FCN is a standard architecture for deep learning models Bagnall u. a. (2017). FCN is a generic architecture usable for any type of data. All the neurons in layer l_i receive the signal from every neuron in the layer l_{i-1} and send their output to every neurons of the layer l_{i+1} with $i \in [1, L]$ (L number of layers in the architecture). The elements of the time series are treated independently from each other. Thus the temporal dimension of the data is ignored with this architecture.

Recurrent neural network (LSTM)

Recurrent neural networks are explicitly devoted to sequence modelling Hochreiter und Schmidhuber (1997). They avoid the long-term dependency vanishing problem using a cells state that is used as internal memory. At each time, the network learns which information to add, forget, and update into the cells state. Based on the cells state, inputs, previously hidden state, the LSTM learn a vector representation (hidden state) of the time series a the current time.

1D Convolutional neural network (CNN)

Motivated by CNN's success in various domains, researchers have also adopted them for time series and sequence modelling. In the context of time series, convolution is a sliding filter over the time series. The time series exhibits only one dimension. Thus this convolution will result in a moving average with a sliding window. Therefore, applying several filters results in learning several discriminative features which are useful for sequence modelling. Besides, the same convolution is used to find the result for all timestamps $t \in [1, T]$ (weight sharing). This is a valuable property, as it enables the model to learn filters that are invariant across the time dimension. 1DCNN for time series is characterized with a causal convolution; It means that the output at time t is convolved only with elements from time t or earlier in the previous layers. This characteristic ensures that the sequence input must have a one-to-one causal relationship in chronological order. The result of convolution can be considered as a time series whose dimension is equal to the number of these filters used. 1DCNN has another layer with pooling operation, which achieves dimension reduction of feature maps while preserving most information B.Zhao u. a. (2017).

Temporal convolution network (TCN)

TCN with dilated convolution is designed to combine simplicity, and long-term memory Bai u. a. (2018). There are many differences with 1DCNN described above. In addition to the causal convolution, the architecture can take a sequence of any length and map it to an output sequence with the same length. To achieve this, zero-padding of length (kernel size - 1) is added. Moreover, the TCN architecture can look very far into the past using a combination

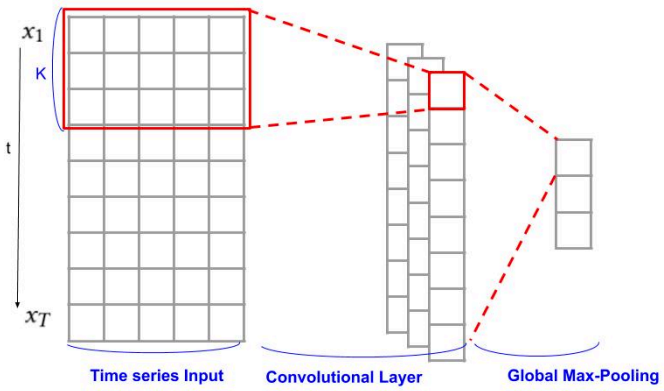


Figure 5.6: 1-D Convolution for Time Series.

of residual layers and dilated convolution. The dilated convolution van den Oord u. a. (2016) enables an exponentially sizeable receptive field using dilation factor d and the filter size k. When using dilated convolutions, we increase d exponentially with the network's depth, allowing a very large history using deep networks.

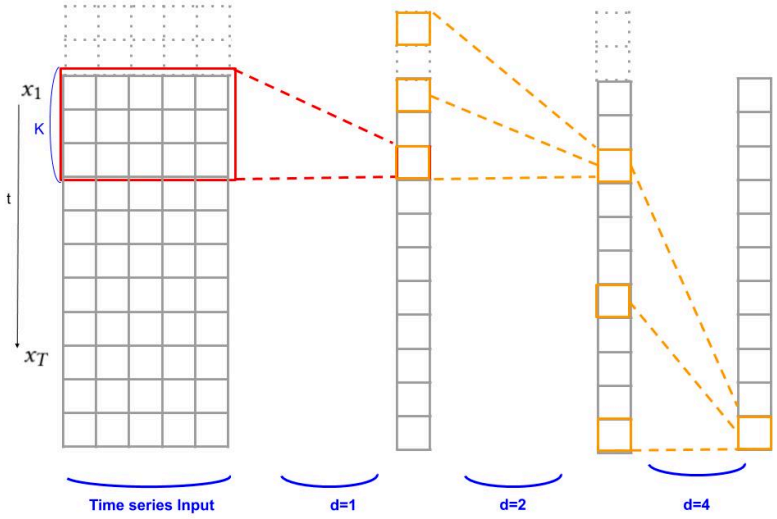


Figure 5.7: Architectural elements in a TCN. (a) A dilated causal convolution with dilation factors d= 1,2,4 (when d=1, regular convolution) and filter size k= 3. The receptive field is able to cover all values from the input sequence.

Discussion

In-vehicle IDS require the model to have good detection results despite strong constraints on computational resources. For FCN, the number of parameters increases exponentially with the input sequence's size, thus increasing the size of the model. CNN and TCN benefit from parallelism; convolutions can be done in parallel since the same filter is used in each layer, unlike in LSTM, where the predictions for later timestamps must wait for their predecessors. LSTM and TCN both include in their architectures a way to deal with the history size, while CNN is limited to

the filter size.

5.3.3 Anomaly detection models

Detecting anomalies raises many difficulties. The major problem is that few or even no anomaly labels are available in the historical data. The goal is to build a system that can face unknown attacks and be able to use the few anomalous examples when available. Therefore, analyzing both One-class and supervised anomaly detection is relevant to design novel IDS. We consider deep learning architectures that are relevant for sequence modelling for each task. We compare those architectures on a one-class feature learning task using an autoencoder and a supervised task using an FCN.

Deep one-class IDS

In this work, the autoencoder aims to reconstruct the input sequence (a multivariate time series). Formally, given a sequence $T = (t_1, ..., t_K)$ where $T_i \in \mathbb{R}^n$ and n is the number of variables, the autoencoder aims at predicting $\hat{T} = (\hat{t}_1, ..., \hat{t}_K)$ at each time (sequence-to-sequence problem). The autoencoder that performs a nonlinear mapping from the current state to its identity decomposes into two parts: the encoder and the decoder. The encoder projects the temporal pattern dependencies and trends of the time series in latent space h. The latent vector is given by $h^i = f(T.W^i + b^i)$, where W^i and b^i respectively denote the weight matrix and bias up to the bottleneck i-th layer. The decoder, considered as the transposed network of the encoder, uses the information of latent space h to reconstruct the input sequence: $\hat{T} = f(h^i.W_d^i + b_d)$. The mean square loss error (MSE) is used to perform an end-to-end learning objective: $L(T, \hat{T}) = \frac{1}{K}\Sigma_i^n(t_i - \hat{t}_i)^2$. At the inference phase, the MSE is used as an anomaly score. The idea is that the autoencoder learnt to reconstruct only the normal data and will obtain a high MSE on the anomaly.

Deep Supervised IDS

Supervised IDS use a FCN to make anomaly prediction from the vector representation of the time series learnt from the sequence modelling level. In this case, we suppose that the training dataset contains labelled attack examples and these attacks from a homogenous class. These requirements are generally reached when we construct a model that detects well-known types of attack. Formally, we assume two classes: Normal (0) and Anomaly (1). The learning set is a collection of pairs $\{(T_1, Y_1), ..., (T_K, Y_K)\}$ where T_i is a multivariate sequence and $Y_i \in \{0, 1\}$ is the corresponding label. The classifier training is performed by minimization of the cross-entropy between the true class and predicted class. Notice that the classes are highly unbalanced, and the anomaly is much smaller than the normal class; the classes are therefore weighted in the cross-entropy in function on their prior. At the inference phase, the MLP returns the probability of anomaly.

5.4 Experiments and results

This section analyses the CAN data by first simulating a CAN network with an injection attack, and we present a more exhaustive public dataset SynCAN. We conduct experiments to assess the impact of the representation on anomaly detection. We use several architectures and discuss the best model choice for an in-vehicle Intrusion detection system, the detection and practical feasibility of the in-vehicle context constraints. [1]

5.4.1 CAN dataset

SynCAN dataset

SynCAN dataset is synthetic data proposed in Hanselmann u. a. (2020); the data set consists of 10 different message IDs, each with varying amounts of signals per ID and different noisy time frequencies. The total number of signals is 20. The data is created in such a way that it is similar to real CAN traffic. It contains physical values (signals), timestamps and IDs. We use a training data set of about 16.5 hours and a test data set of about 7.5 hours of CAN traffic. The data set is available at https://github.com/etas/SynCAN Hanselmann u. a. (2020). We evaluate our model on the following attack types:

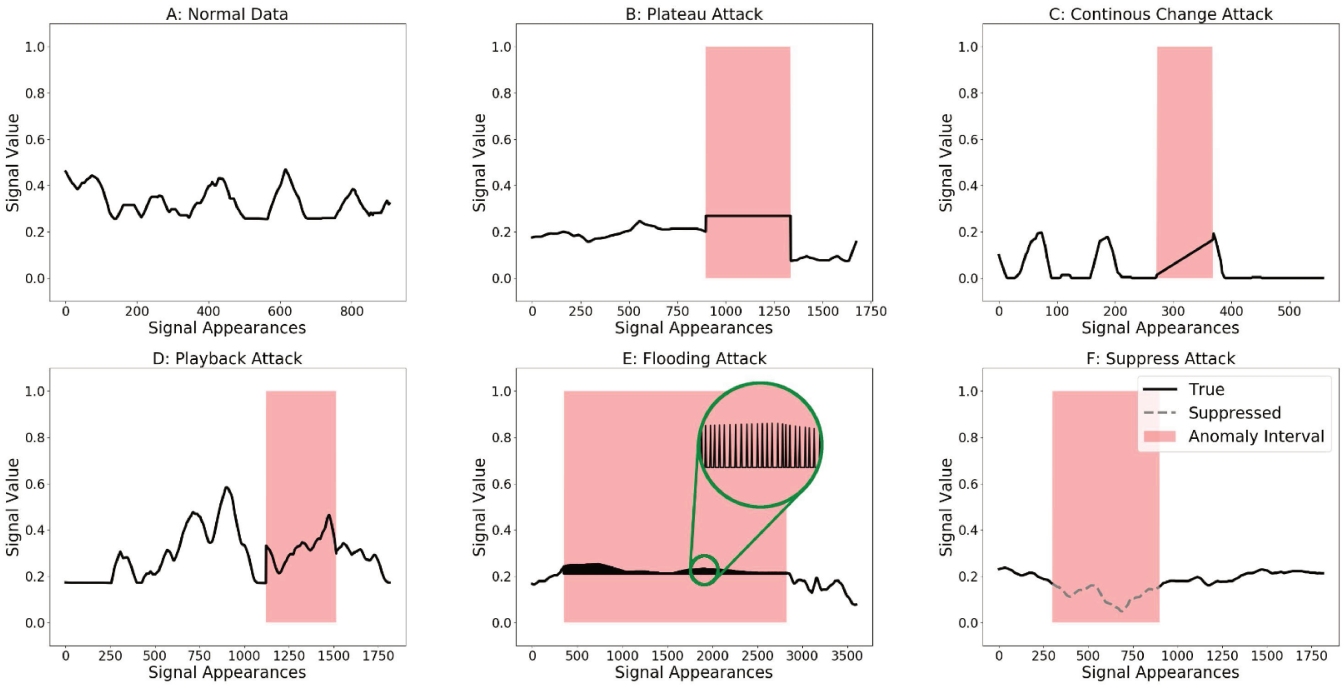


Figure 5.8: Visualization of CAN data with different attack types on short time intervals. The flooding attack contains high frequency anomalies between its real signal values. In the suppress attack the dotted line represents real signal values that are not transmitted onto the CAN bus. Hanselmann u. a. (2020)

Plateau attack: A signal is overwritten to a constant value over a period of time. Smaller jumps or freezing might

[1] The implementations and details are be available at : https://github.com/anonymeEG/Deep-Learning-4-IDS.

only be detected if we consider a set of signals with some correlation between them.

Continuous change attack: A signal is overwritten so that it slowly drifts away from its true value. This assumes that the attacker wishes to set a signal to a concrete value while trying to fool the IDS with realistic small changes in the signal.

Playback attack: A signal value is overwritten over a period of time with a recorded time series of values of that signal. The attacker hopes to trick the IDS by sending completely real looking signal values of a different traffic situation.

Suppress attack: The attacker prevents an ECU from sending messages, for example, by turning it off. This kind of attack means that some particular ID messages do not appear in the CAN traffic for a period of time.

Flooding attack: The attacker sends messages of a particular ID with high frequency to the CAN bus. This attack is easier to perform in practice than the ones mentioned above since the attacker does not need to control an ECU. It only requires sending additional messages to the CAN bus to "overwrite" the real message values.

Overall, most attack in the test data last 4s (See Fig 5.8). Also, we can categorise the attack impact as flow or payload based. Flooding and suppress are visible when we look at the flow behaviour, while plateau, playback, and continuous attack types are more visible when we look at the payload signal values. The below section will give the matrix construction's different parameters that yield a representation allowing the sequence modelling to consider both flow and payload information.

Occurrence matrix settings

We set five seconds as an estimated time-frame for the intrusion detection system to monitor the vehicle. Thus, the sampling window is fixed to $\Delta = 50ms$, and each sequence is composed of $K = 100$ consecutive elements. From a general standpoint, K and Δ are hyperparameters which depend on the domain expert requirement (the maximum memory size, forensic analysis and safety protocol to enable the prevention actions). The feature matrix size is (100*30) where 30 is the sum of 20 signal features and 10 occurrence features. We scale the data between $[0, 1]$ using min-max normalization.

A sequence is labelled normal if all elements in the sequence are normal. If a sequence contains at least one anomaly, the sequence is considered as an anomaly. We are trying to assess whether the IDS can detect the attacks even at an early stage. SynCAN database is a collection of $\sim 2'000'000$ normal sequences. 70% of them are used for training and 10% for validation. The last 20% are mixed with anomalous sequences to build the test data. We have 5 test databases, one per attack, made of 70% normal examples and 30% anomalous examples.

5.4.2 Models training and Results

We use four different architectures (FNN, CNN, TCN and LSTM)with two experiment settings: One-class anomaly detection and supervised anomaly detection.

We have trained the models on 500 epochs for all architecture, with a batch size of 100. We used adam Kingma u. a. (2014) as the optimizer for the gradient descent with learning rate decay when a metric has stopped improving. Models often benefit from reducing the learning rate by a factor of 0.2 after 100 epochs. And we shuffle the data in each epoch. For these experiments, we have used Keras and TensorFlow (Chollet u. a., 2015; Abadi u. a., 2015) as the main deep learning framework. We used Nvidia Tesla M40 and Nvidia Tesla P100 GPUs.

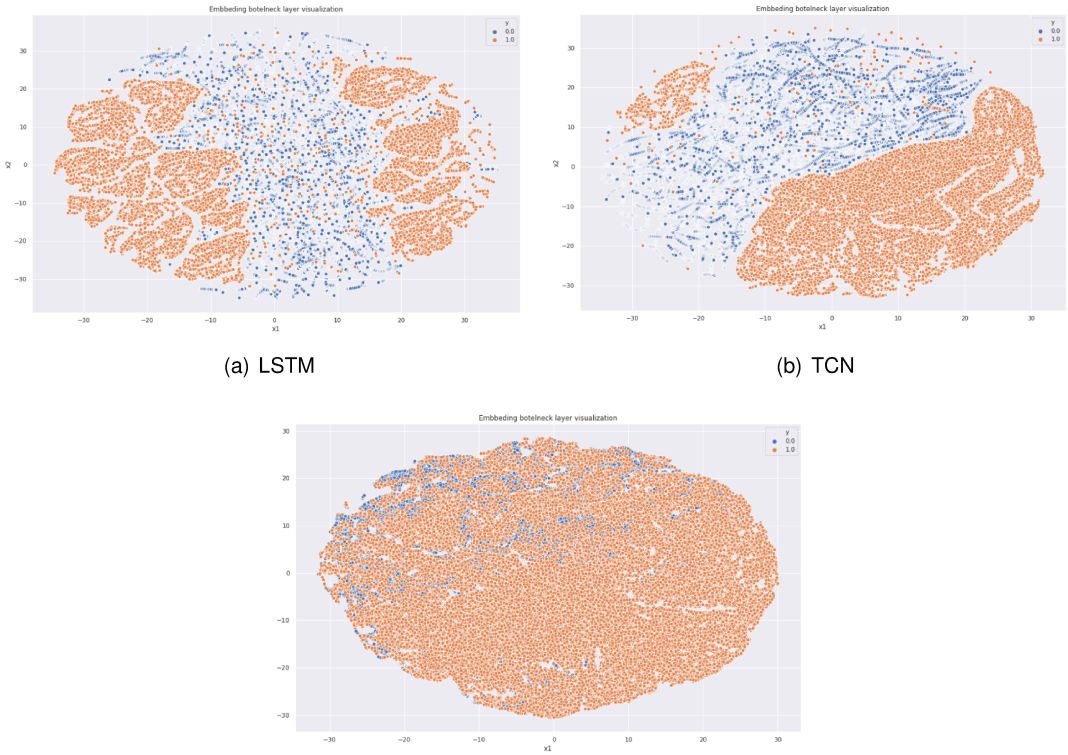
We evaluate the model's performance with F-measure, Precision and Recall metrics. Those metrics are related to basic metric, True positives (TP) and true negative (TN), both measure the number of attack and normal example respectively that were correctly predicted by the model. False Positive and False-negative represent the proportion of attack and normal example that were incorrectly predicted. So, Precision $Prc = TP/(TP + FP)$ and Recall $Rec = TP/(TP + FN)$, respectively, describe the percentage of correctly identified attacks versus all predicted attacks and all actual existing attacks. F-measure is the weighted average of Precision and Recall. $F_1 = (Prc + Rec)/(Prc + Rec)$.

In Table 5.1, we show the metrics on the One-class task using autoencoders with different architectures. All architectures show excellent performances for all types of attack. TCN is slightly better on most attack cases and comparable with LSTM on *Plateau* attack. Notice that on the *Suppress* attack, the models perform worse than on the other attacks, while the CNN collapses with a lot of false positive. It shows that *Suppress* attack is unobtrusive. Moreover, in the representation matrix M, there is no explicit mention of the missing values. Nevertheless, TCN and LSTM still have good results. Thus they can implicitly retrieve the information in the learning stage from the representation matrix.

	TCN			LSTM			CNN			FCN		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Continues	**0.997**	**0.991**	**0.994**	0.991	0.988	0.990	0.996	0.988	0.992	0.993	0.978	0.985
Plateau	0.995	0.984	0.990	**0.996**	**0.985**	**0.991**	0.993	0.979	0.986	0.990	0.981	0.987
Suppress	**0.986**	**0.957**	**0.971**	0.984	0.954	0.969	0.951	0.554	0.700	0.951	0.862	0.904
Flooding	0.995	0.986	0.991	0.996	0.988	0.991	**0.996**	**0.989**	**0.992**	0.996	0.988	0.991
Playback	**0.996**	**0.989**	**0.992**	0.996	0.986	0.991	0.994	0.989	0.991	0.995	0.988	0.991

Table 5.1: Autoencoder based architectures results

In Fig 5.9 we show the visualization of the latent representation using the t-Distributed Stochastic Neighbor Embedding (t-SNE) technique for dimensionality reduction that is particularly well suited for the visualization of high-dimensional datasets van der Maaten und Hinton (2008). The latent space contains a compressed representation of the CAN data input (Occurrence matrix), which is the only information that the decoder can use to reconstruct



(a) LSTM

(b) TCN

(c) CNN

Figure 5.9: Visualizing the encoder bottleneck layer (Latent space) using t-SNE.

the entry as faithfully as possible. To this end, the network must learn to extract the most relevant features from the bottleneck learned via sequence modelling architectures (a) LSTM, (b)TCN, (c)CNN. We can clearly see that the visualization reflects the results obtained in the Table 5.2, the normal examples are isolated from the attacks examples in both Fig 5.9(a) and Fig 5.9(b) while in Fig 5.9(c) the attacks and normal example are indistinguishable.

In Table 5.2, we show the experiments on the supervised task (Binary classifier). TCN is still slightly better than the others and close to CNN. Notice that for LSTM architecture, the results are not as good as in the One-class task. Indeed, we have reduced the rate of normal data in training set to rebalance the data and help the model not learn from the normal features only. LSTMs are more data-hungry than CNN and TCN. It shows that TCN needs less data than LSTM for CAN data modelling.

	TCN			CNN			LSTM			FCN		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Continues	**0.996**	**0.986**	0.990	0.995	0.991	0.995	0.959	0.933	0.948	0.89	1.00	0.94
Plateau	**0.997**	**0.998**	0.997	0.991	0.953	0.971	0.984	0.953	0.968	0.87	1.00	0.93
Suppress	**0.999**	**0.999**	**0.999**	0.998	0.998	0.999	0.992	0.999	0.995	0.86	1.00	0.92
Flooding	**0.999**	**0.999**	**0.999**	0.999	0.999	0.999	0.999	0.999	0.999	0.87	1.00	0.93
Playback	**0.997**	0.988	0.992	0.997	0.995	**0.995**	0.994	0.989	0.991	0.89	1.00	0.94

Table 5.2: Classification using the occurrence matrix representation

In Table 5.3, we conduct the same experiment, but we eliminate the occurrence features from the representation

matrix M. We notice for the *Flooding* attack, the performances of all the models decreases dramatically. Indeed, this attack impacts the CAN data flow, and this information is encoded through the occurrences in the matrix. We also observe that the performances are slightly worse on the *Playback* and *Plateau* attacks. Therefore, payload-based attacks are also easier to detect when the occurrence features are present in the matrix. Hence full matrix with both signal features and occurrence features contributes to the detection of both payload and flow-based attacks.

	TCN			CNN			LSTM		
	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Continues	0.995	0.994	**0.994**	0.983	0.990	0.986	0.945	0.957	0.950
Plateau	0.973	0.978	0.975	0.995	0.998	**0.996**	0.959	0.984	0.971
Suppress	0.986	0.996	0.990	0.990	0.971	0.980	0.939	0.969	0.953
Flooding	0.985	0.978	0.981	0.971	0.915	0.928	0.927	0.972	0.913
Playback	0.992	**0.998**	0.994	0.992	0.772	0.868	0.935	0.924	0.929

Table 5.3: Classification using the standard sampling without occurrence features

Finally, in Table 5.4, we have compared the models in terms of training time and size of parameters. The latter reflects the memory needed by the IDS to work. Remind that the IDS are embedded in-vehicle where memory and computational resources are limited. TCN is good both in terms of training time and model size. TCN benefits from filters shared weight, so it dramatically reduces the number of parameters. When the size of the input data is increasing, the number of parameters does not explode exponentially. Unlike LSTM, TCN convolutions can be done in parallel since the same filter is used in the layer. Therefore, even though the series is long in both the training and inference phase, it can be processed as a whole, whereas with LSTM, they must be processed sequentially.

Models	Training time	Number of parameters	model size (32-bits floats)
FCN	8022s	75238	0,3 MB
CNN	10011s	9518	0.03 MB
TCN	7969s	3822	0.01 MB
LSTM	92714s	2920	0.01 MB

Table 5.4: models Computational resources

5.5 Conclusion

In this chapter, we discussed our second contribution towards developing an in-vehicle Intrusion Detection System. We introduce a novel in-vehicle intrusion detection system based on an extensive series of experiments that validate the different levels of the system: 1) At the data level, we use a representation matrix to structure the CAN data information that groups both flow and payload information. 2) At the sequence modelling level, we use a TCN architecture since we have shown that it performs well concerning the detection metrics and computational resources consumption. 3) At the detection level, we jointly use a classifier and an autoencoder so that the IDS can deal with both known and unknown attacks. Notice that our results were established by using the SynCAN Dataset, which is

the only available public dataset as far as we know.

The in-vehicle system has many components, and we have implicitly assumed that the monitoring of the data was centralized. Nonetheless, in new secured in-vehicle architectures, the system's parts are isolated, so the CAN data topology changes, and we need to think about a distributed framework for IDS. Another important issue concerns deep learning models' compression to fit the in-vehicle system's embedded capacity better.

Chapter 6

Distributed anomaly detection based in-vehicle intrusion detection system

In this chapter, we focus on the in-vehicle network infrastructure and deployment. Future in-vehicle (autonomous vehicles) network architectures will consider many aspects of modern network security by design. The general system contains many sub-systems related to different tasks with specific functional priorities and dedicated security mechanisms. For example, a decision pilot sub-system is responsible for the functions related to autonomous driving. The infotainment sub-system is related to different operations associated with passengers' entertainment; both systems are parts of the global system. Still, for security reason, both sub-systems are isolated, and no communication is possible between the sub-systems as the infotainment, since infotainment sub-system is more exposed to external communication (internet) (see Chapter 2 Section 2.1.1). In this work, we propose a Distributed Anomaly Detection IDS (DAD) using a deep learning model that fits the in-vehicle network architecture. DAD aims to model the complex correlations among different views (sub-systems) while adopting the same isolation constraint on the sub-systems. On top of that, we introduce a new optimisation scheme that lowers both the computational inference time and the IDS's communication overhead.

6.1 Introduction

Future In-vehicle network architecture is composed of different subsystems (ECUs). Each subsystem is responsible for specific services that ensure the autonomous functioning of the vehicle. For functional and security reason, separate subsystems are isolated, forming a hierarchical architecture of the whole system. In that context, data is represented with different views and can include multiple modalities or various features. These views may be obtained from multiple sources or different feature subsets. In this work, each subsystem contains a probe (S) that

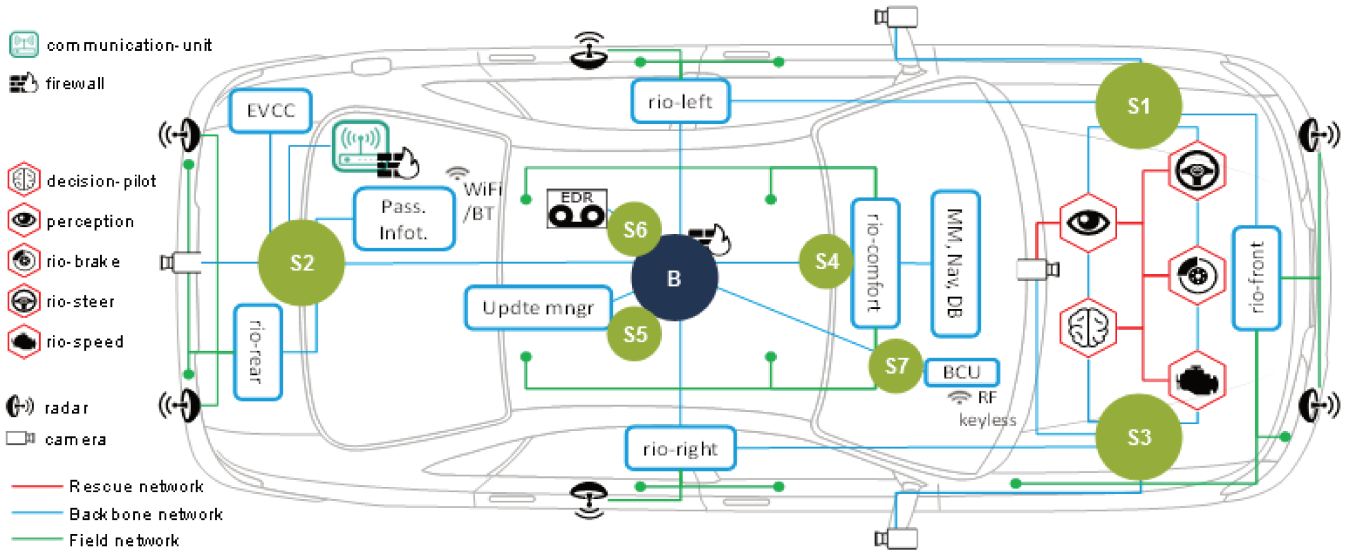


Figure 6.1: Schema representing the different probes that monitor various subsystems of an autonomous vehicle architecture network.

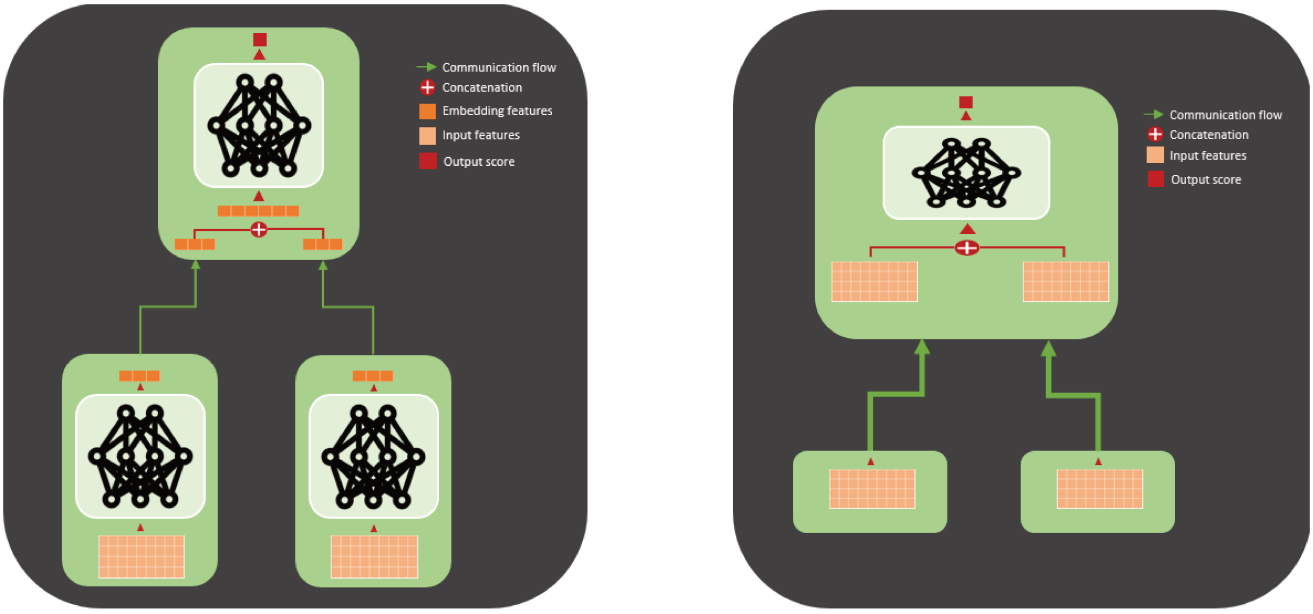
monitors its local environment (CAN network) and communicate only with the central probe bastion (B) (see Fig 6.1).

In the context of a distributed intrusion detection system with its inner architectural characteristics, we define two strategies to build a Deep Anomaly Detection model for IDS (See Fig 6.2).

1. Centralized anomaly detection modelling: In this case, the computational resources are located only on the bastion (See Fig 6.2(b)), and the model is embedded on the bastion. Each probe sends its data information to the bastion. So, the bastion aggregates this information and predict the state of the system based on one model.

2. Distributed Anomaly Detection Modelling: (As described in Fig 6.2(a)), a common strategy is to learn a joint representation coupled between multiple views at a higher level after learning several layers of view-specific features related to the specific probe. Each probe's particular view is represented with a lower dimension vector (feature vector). Those feature vectors jointly leverage the abundant and complementary information from multiple views. From a neural network modelling perspective, multi-view representation learning first learns the respective mid-level features for each view (probe) using a sub-neural network. The bastion then integrates the complementary knowledge of different views to comprehensively represent the initial data into a single and compact representation.

This work focuses on distributed anomaly detection using deep learning represented in Fig 6.2(a). We propose a Distributed Anomaly Detection framework (DAD) intrusion detection system using a multi-view deep learning architecture. The proposed deep learning architecture respects the in-vehicle network topology and constraints (porobes communicate only with the bastion). It aims to classify the network's current state by learning on local



(a) Distributed Anomaly Detection Modelling

(b) Centralized anomaly detection modelling

Figure 6.2: (a) Represents DAD modelling neural network architectures. The sub-networks are embedded into probes and learn to capture the local pattern. They send a feature vector to the upper sub-network located in the bastion that will take the joint representation that leverages different views' information complementarity. (b) It is a centralized model based on a single view. The different probes send their raw data to the bastion that concatenates them into a single view.

patterns related to each view. Thus, the subspace learning-based approaches (multi-view representation learning) aim to obtain a joint feature vector subspace shared by multiple views by assuming that it represents the input data information. The received feature vector's dimensionality is lower than the input view, so subspace learning effectively reduces the "curse of dimensionality". Given this subspace, it is straightforward to conduct subsequent tasks, such as classification for anomaly detection. Furthermore, DAD reduces the communication overhead induced by integrating an intrusion detection system on top of the existing in-vehicle network functionalities. Indeed, in-vehicle systems, in general, have low computational power, and any additional computational resources (processing and communication overhead) can be translated into a high cost for automotive manufacturers. We conduct detailed experiments to assess the relationship between different sub-networks predictions and their relation in representation learning and accuracy of classification on different attacks types using SynCAN dataset (Introduced in Chapter 4 Section 5.4.1). The output predictions value are processed by a novel abnormal score mechanism with new metrics, including the communication overhead optimisation. If a normal condition occurs, the probes remain silent, and no further processing is needed. DAD IDS is the first practical in-vehicle intrusion detection that fits the distributed network architecture while introducing the notion of communication overhead optimisation to the best of our knowledge.

6.2 Related works

Multi-view learning and multi-modal Deep learning architectures have been concerned in many anomaly detections problems, especially in network environments forming many swarms like VANet's, cloud computing and edge computing areas. In those cases, most data is collected from different sources, or different features extractors Xu u. a. (2013); Peng u. a. (2018); Ding u. a. (2019); Marcos Alvarez u. a. (2013); Ji u. a. (2019). In other words, data instances are usually depicted by heterogeneous feature spaces in the form of multiple views. Several multi-view learning approaches can be considered to develop an anomaly detection model based IDS. Multi-view representation learning is concerned with learning representations (or features) of the multi-view data that facilitate extracting readily useful information when developing prediction models. Data from different views usually contains complementary information, and multi-view representation learning exploits this point to learn more comprehensive representations than single-view learning methods Li u. a. (2019); Xu u. a. (2013). There has been increasing research applied to multi-view learning using Deep Learning, deep architecture based methods including multi-modal deep Boltzmann machines Srivastava und Salakhutdinov (2014), multi-modal deep autoencoders Feng u. a. (2014); Wang u. a. (2015), and multi-modal recurrent neural networks Donahue u. a. (2017). In the existing literature, most in-vehicle IDS neglect the inherent multi-view property of data due to the lack of IDS problem modelisation concerning future in-vehicle architectures. The lack of datasets related to it is mainly the reason. Thus, building and IDS's challenges and constraints for future in-vehicle architectures are not implemented in most in-vehicle IDS literature. Few works tackle building an in-vehicle IDS that monitors the CAN bus network by proposing a structure of distributed anomaly detection system. In Wang u. a. (2018) the author propose a distributed IDS based on hierarchical temporal memory (HTM). The input at each detector in the sequence is the bits from the packet's data field related to each ID. Then the model using the HTM algorithm learns to predict the next data field of each ID. An overall score within a time window for the full input sequence groups the different IDs scores. InHanselmann u. a. (2020) the authors also tackled the problem in the same manners in terms of distributed modelling, where the data input modalities are sequences related to a specific ID. The authors proposed multi-view architecture based on independent recurrent neural networks (LSTM for each ID that gets sequence associated with this ID as inputs). The joint latent vector is fed into a subnetwork of consecutive linear layers in an autoencoder setting. At each time step, this subnetwork's task is to reconstruct the signal values of each possible input message solely based on the current joint latent vector. The drawback in the IDs based distribution is that both propose several independent subnetwork equal to the number of IDs. In other terms, those architectures and algorithms don't consider the restrained resources available in the in-vehicle systems. The communication overhead is augmented with a number of signals and processing equal to the number of IDs available on CAN data. In this work, we formulate the problem related to this new type of in-vehicle architectures and propose a general framework that can be extended to fit autonomous vehicles network systems.

6.3 Method and modelling

6.3.1 Problem Statement

We consider a distributed architecture for network intrusion detection system. Firstly, we assume that we have p probes $(S_1, S_2, ..., S_p)$ that monitors the vehicle's different subnetworks (CAN bus communication between different ECUs). Each probe S_i hosts a local anomaly detector D_i. The input of detector D_i is a multivariate time series T_i. The aim of D_i is to distinguish between the normal pattern and the attack pattern based on the features extracted from T_i only. Formally $D_i(T_i) \approx \mathbb{P}(y_i = 1|T_i)$ where $y_i \in [0, 1]$ (with 1 for normal example and 0 for anomalous examples).

The probes are not allowed to exchange information between them. They exchange information only with the bastion B. We want to limit this exchange of information as much as possible to reduce the communication overhead. To achieve this goal, we use two levers. On one hand, the detectors do not send information at each step of time: it assesses the probability of an attack $D_i(T_i)$ and raises an intrusion alarm $e_i \in \{0, 1\}$ if this value is larger than a given threshold τ_i. Formally,

$$e_i = \begin{cases} 0, & \text{if } D_i(x_i) \geq \tau_i \\ 1 \end{cases} \tag{6.1}$$

If no probe raises the alarm, then no information is sent to the bastion by any probe. On the other hand, if only one of the probes raises the alarm, the bastion asks all the probes to transmit information about their status and local data. This is the second lever where we act: as we shall see below, the detectors are based on deep learning and designed to provide a condensed (summarized) representation of the data using a function G_i. Therefore, the detector D_i does not transmit raw input data T_i but a condensed representation denoted $v_i, v_i = G_i(T_i)$. The bastion hosts an anomaly detector D_B that groups the different representation v_i to decide on the global system behaviour $D_B(V) \approx \mathbb{P}(y_B = 1|V)$.

This allows reducing the communication overhead dramatically. To quantify this, let us introduce the following,

communication cost function ζ: Given any vector W , let $|W|$ denote the size of the vector; then

$$\zeta(W) = k|W| + H \tag{6.2}$$

where H is the incompressible cost of initiating a message sending and k is a parameter that allows adjusting the importance of both criteria.

Now, with respect to an input series of raw data T_i, let N be the number of data points in a given dataset; N is the number of messages that would transit through the network to the bastion B in the absence of local detector D_i (Centralized model see Fig 6.2(b)). In the presence of detector D_i, the number of such messages is dramatically

reduced, so let N_e be the number of messages sent from D_i to the bastion B. Moreover, the transmitted information is v_i which is much lighter than T_i, so the communication gain of our techniques is the following:

$$gain = \sum_{1}^{N} \zeta(T_i) - \sum_{1}^{N_e} \zeta(v_i) \tag{6.3}$$

We note that the choice of the threshold τ_i impacts the detection precision of the model. An over optimisation is when the probes don't send the features vector representation v_i to the bastion, and at the same time, an attack occurs. In this case, the communication optimisation also reduces the precision of detection and generates an error. The aim is to maximise the precision of detection and reduces the communication cost. The gain represents the optimisation rate accomplished without committing an error.

Altogether, these severe communication requirements of having negligible overhead induced by the IDS, reserving ample resources for the vehicle's intended network services. Below, we present the architecture and the different steps of modelling a distribute anomaly detection IDS (DAD) subject to design a practical solution for the aforementioned problem.

6.3.2 Proposed model

The overall goal is to build a model that detects attacks in a car when they happen. The model must respect and fit the in-vehicle network's distributed architecture and, more precisely, reduce the communication overhead brought by the detection process. The framework of our approach is summarized in Fig 6.3. The first Block is composed of a separate sub-network based on the Temporal Convolutional Network to learn the local feature representation of each input sequence. A Fully Connected Network classifier takes the joint feature representation yielded by the sub-network models to return the global network system's state.

We propose a hierarchical Distributed intrusion detection system based on multi-view deep learning architecture to respond to the problem mentioned above (Section 6.3.1). The model has multiple inputs sequences T_i related to each probe S_i. The model returns multiple outputs $\{\hat{y}_i\}_{i=1}^{P}$ and $\hat{y}_B$. $\hat{y}_i = D_i(T_i)$ represents the local normality probability for each probe, and $\hat{y}_B = D_B(V)$ represents the bastion probability of the normality of the whole system. We denote y_d the final decision based on the combination of $(\{\hat{y}_i\}_{i=1}^{P}, \hat{y}_B, e_i)$.

The model captures each probe S_i local pattern as a smaller feature vector representation of the input through a sequence modelling transformation G_i. Those Features vectors v_i are sent to the bastion B to predict the global system state. We want to optimise the overhead communication induced by embedding our IDS in the in-vehicle network system. Thus, using a distributed network reduces the size of the information sent from the porobes to the bastion by sending the feature vector v_i representation alternatively of data input sequence T_i. The second point is to reduce the communication rate between the probes and bastion. The idea is to prevent sending the features

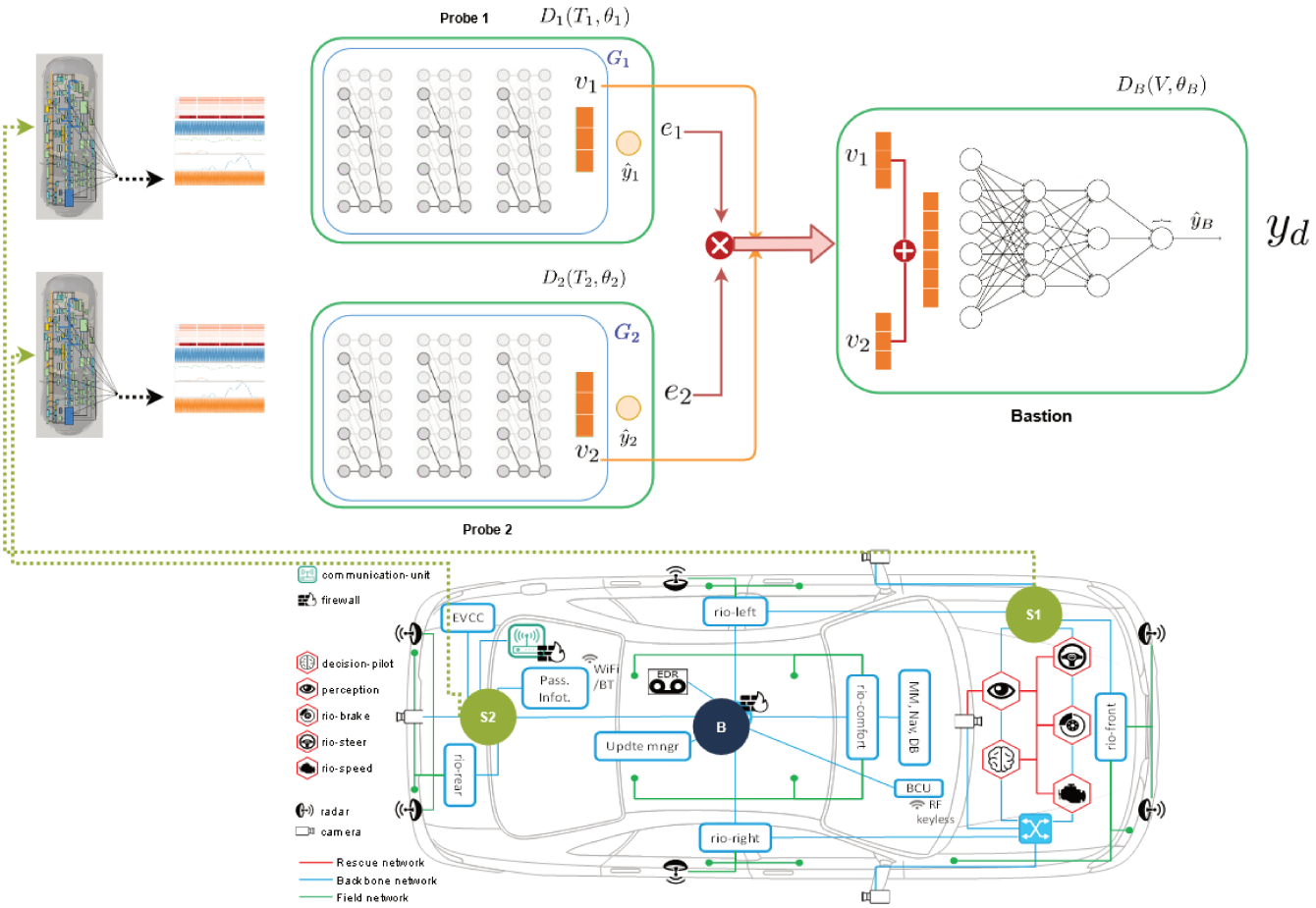


Figure 6.3: The framework of the proposed model. We consider two-model classification, but the proposed framework is not limited to two-model. The raw feature consists of the matrix occurrence of the encoded multivariate binary sequence. Both of them are fed to the models D_1f and D_2 for sequence feature extraction and local anomaly prediction. The D_B model receive both v_1 and v_2 for a global anomaly detection prediction y_d .

vector representation when the probes are certain of their local pattern's normality. To this end, we introduce a new target e_i (see Eq 6.1) for each probe to decide whether to send the feature vector v_i to the bastion based on threshold optimisation that we will explain below.

To learn the model D_i we use (Temporal Convolutional Network) architecture for the sequence modelling to achieve the local classification task.

The condensed representation v_i is the vector representing the embedding feature given by $v_i = G_i(T_i)$, where G_i the sequence modelling transformation function represented with the TCN layers (layers of the embedding feature before softmax layer). The sub-network D_B is a fully connected neural network that takes V, $V : (v_1, ..., v_p)$ as an input, by regrouping all the embedding features representing the local patterns, the bastion B returns the probability y_B on global features. The goal of this architecture is to learn classifiers $D_i(T_i, \theta_i)$ and the bastion classifier $D_B(V, \theta_B)$ that yields the prediction vector $\hat{Y}, \hat{Y} : (\{\hat{y}_i\}_{i=1}^{P}, y_B)$ simultaneously by optimising the parameters $\theta = (\theta_1, ..., \theta_p, \theta_B)$.

Model learning

Since dependencies among local labels and models play an important role in extracting hierarchical dependencies, we privilege the local classification and global classification, meaning that each sub-network model D_i learn a local feature representation relative to each probe. Those representations will then be jointly fed to the upper sub-network classifier D_B to learn about a global pattern based on the join feature representation vector. Hence, giving a second assessment of the network's general state with the overall network information's abstraction. we define the objective function as Eq 6.4:

$$L_{DAD} = \sum_{i}^{p} L_{S_i} + L_{S_B} \tag{6.4}$$

The first terms is the loss functions of model D_i. The next term is the global classifier's D_B loss function, which analyses dependencies among the feature vectors v_i to learn the network system's global pattern. Both terms represented with the *binary cross entropy* loss function (See Eq 6.5, Eq 6.6). We note that Y represent the true label for a given training sample (T, Y).

$$L_{S_i}(T_i, Y) = -Ylog(D_i(T_i))) - (1 - Y)log(1 - D_i(T_i))) \tag{6.5}$$

$$L_{S_B}(V, Y) = -Ylog(D_B(V))) - (1 - Y)log(1 - D_B(V))) \tag{6.6}$$

The proposed framework leverages both local label dependencies, and models relations throw feature representation to facilitate the learning process. It constructs deep network representation and classifier for each view and can make a single view prediction for specific local view. For the model learning, we propose to combine the sub-networks D_i and the global classifiers D_B together. The backpropagation is used to train the sub-networks and classifiers jointly.

Threshold optimisation

After the learning phase, the proposed approach has multiple output prediction values. For a given test samples the following probabilities are yield $preds = (\{\hat{y}_i\}_{i=1}^{P}, \hat{y_B})$

The intrusion detection final decision based on the DAD model is not directly based on the values returned in $preds$. Indeed, we add another step that we call the optimization step. As explained above, we aim to make DAD predictions more economical in terms of communication overhead. To this end, we deduce e_i from $\hat{y}_i$ according to

Eq 6.1, based on the threshold of certainty τ_i where $e_i \in [0, 1]$ (1 for sending the message, 0 for preventing sending the message). Each probe S_i send its related embedding features v_i when its model D_i yield a score $\hat{y_i}$ lower than the threshold τ_i, that defines how much the model is certain about the normality of the local pattern in the probe S_i given a sequence T_i.

We define $y_d = f(E, \hat{y}_b)$ the final anomaly score Eq 6.7, where $E : (e_1, ..., e_p)$.

$$y_d = \begin{cases} 1, & \text{if } e_i = 0, \quad \forall e_i \in E \\ \hat{y}_b \end{cases} \tag{6.7}$$

y_d represents the final anomaly score yield by DAD, if $\forall e_i \in E, e_i = 0$, $y_d = 1$ it means that all the probes flagged an affirmation of the normal state of the system and no more processing is needed, so the probability score of y_d is set to 1. In that case, two optimizations are done, the first is the processing of the model D_b, and the second is the communication of the different representation vectors v_i. On the other hand, if one probe is not certain about its local state's normality, $\exists e_i \in E, e_i = 1$, the bastion is requested to launch the model D_B and all v_i are sent. In that case, the only notable optimization is the size of v_i compared to T_i in a centralized model.

6.4 Experimentation and results

This section presents the different experiments setting and results concerning the workflow used to build the DAD framework. We start with adapting the SynCAN data for a multi-view input. The second part is to train the sub-networks jointly. The third part is to calibrate the different thresholds to obtain good accuracy detection while reducing the communication overhead.

6.4.1 Data and experiment setup

Dataset preparation:

We used the SynCAN dataset that we presented in the previous chapter (Section 5.4.1). As the SynCAN dataset's capture was mainly done buy indexing only the different messages' IDs, we don't have the information of the ECUs related to each ID. Thus we cannot split to different view according to a semantic locality pattern as shown in the architecture presented in Fig 6.3. We split the IDs according to each ID's frequency, so we obtain a balanced rate number of messages at each probe. We obtain two views referring to two probes S_1 and S_2, and each probe monitors five IDs. The following step is to create the sequence matrix for T_1 and T_2. We obtain two datasets with an input size is (100*12, 100*8) where 100 is the length of the obtained sequence and 8, 12 are the number of signals monitored at each probe S_1 and S_2 respectively.

For the anomalous examples labelling we choose to set $y_1 = y_2 = y_b = 0$ if $y_1 = 0$ or $y_2 = 0$. This labelling aims

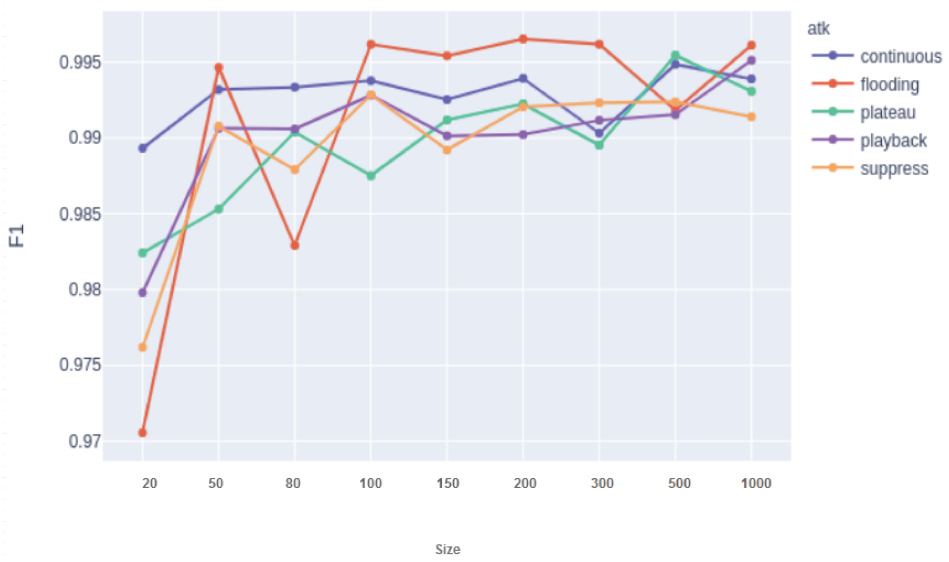


Figure 6.4: Visualization of the F1 score with different layers after the joint representation (First layer of the classifier D_B).

to encourage the sub-network to learn more about the certainty of sending the message by backpropagating the error relative to the all system state while having only the information on the local sequence. We split the dataset into 80% training set and 20% for testing.

Training settings

We train our DAD model for 100 epochs, and we set the batch size to 100. We used the ADAM optimizer for the gradient descent, and we put the learning rate to 0.0001 (decayed by a factor of 0.1 after 50 epochs). As for the TCN parameter, we use one dilated factor d=2 into the 1DTCN. We use Keras for the training under Nvidia Tesla M40 GPUs. The architecture and parameter size is shown in Fig 6.5. In figure Fig 6.4, we show the tunning of the first layer of D_B that connect to the joint feature vector.We set the size of this layer to 100 as it shows the best trade-off between the number of parameters and the model's performance in terms of F1 measure.

```
Layer (type)                 Output Shape              Param #     Connected to
==================================================================================================
left_input (InputLayer)      (None, 200, 12)           0
__________________________________________________________________________________________________
right_input (InputLayer)     (None, 200, 8)            0
__________________________________________________________________________________________________
left_conv1 (Conv1D)          (None, 200, 12)           1740        left_input[0][0]
__________________________________________________________________________________________________
right_conv1 (Conv1D)         (None, 200, 12)           1164        right_input[0][0]
__________________________________________________________________________________________________
left_pool1 (AveragePooling1D)(None, 33, 12)            0           left_conv1[0][0]
__________________________________________________________________________________________________
right_pool1 (AveragePooling1D)(None, 33, 12)           0           right_conv1[0][0]
__________________________________________________________________________________________________
left_conv2 (Conv1D)          (None, 33, 32)            2336        left_pool1[0][0]
__________________________________________________________________________________________________
right_conv2 (Conv1D)         (None, 33, 32)            2336        right_pool1[0][0]
__________________________________________________________________________________________________
left_pool2 (AveragePooling1D)(None, 5, 32)             0           left_conv2[0][0]
__________________________________________________________________________________________________
right_pool2 (AveragePooling1D)(None, 5, 32)            0           right_conv2[0][0]
__________________________________________________________________________________________________
left_feature (Flatten)       (None, 160)               0           left_pool2[0][0]
__________________________________________________________________________________________________
right_features (Flatten)     (None, 160)               0           right_pool2[0][0]
__________________________________________________________________________________________________
concatenate_3 (Concatenate)  (None, 320)               0           left_feature[0][0]
                                                                   right_features[0][0]
__________________________________________________________________________________________________
dense_5 (Dense)              (None, 100)               32100       concatenate_3[0][0]
__________________________________________________________________________________________________
dense_6 (Dense)              (None, 10)                1010        dense_5[0][0]
__________________________________________________________________________________________________
main_output (Dense)          (None, 1)                 11          dense_6[0][0]
__________________________________________________________________________________________________
left_output (Dense)          (None, 1)                 161         left_feature[0][0]
__________________________________________________________________________________________________
right_output (Dense)         (None, 1)                 161         right_features[0][0]
==================================================================================================
Total params: 41,019
Trainable params: 41,019
Non-trainable params: 0
```

Figure 6.5: Illustration of the different parameters of DAD architecture

6.4.2 Results

We start by showing and assessing the results concerning the accuracy of detecting anomalies at each probe. We evaluate DAD model performance on the different probes using F1-measure, Precision and Recall metrics; we show the results in Table 6.1.

From the Table 6.1 we observe a clear gap between the results on the global classifier D_B than D_1 and D_2. First, the observation of the results on y_B concludes that hierarchical learning from the feature vector representation v_1, v_2 obtained from D_1 and D_d is effective. The sub-network sequence modelling task is well learned, and it preserves the information in the feature vector V with a reduced dimension where $|V| < |T_1| + |T_2|$, so D_B Discriminator can differentiate between normal traffic and attacks. On the other hand, we can explain the under-performance of the sub-network results D_1 and D_2 compared to D_B with the lack of general view on the system and more correlation dependencies are needed. Thus, local pattern modelling is not sufficient to get good results in detecting specific attacks.

Table 6.1: Performance on DAD model. We evaluate the model on different attack classification. The *tresh* column is the default threshold set to discretize the probability score returned by D_1, D_2, D_B. CNNeq refers to the use of 1DTCN architecture with dilated parameters d=2, and for the labelling setting equal for eq $y_1 = y_2 = y_B$.

	model	atk	probe	Precision	Recall	F1	TP	TN	Accuracy	tresh
0	CNNeq	continuous	B	0.997025	0.994301	0.995662	0.994301	0.998913	0.997676	0.5
1	CNNeq	continuous	S1	0.974143	0.894067	0.932389	0.894067	0.991302	0.965222	0.5
2	CNNeq	continuous	S2	0.953365	0.825059	0.884583	0.825059	0.985207	0.942253	0.5
3	CNNeq	flooding	B	0.996559	0.997767	0.997162	0.997767	0.998745	0.998484	0.5
4	CNNeq	flooding	S1	0.957714	0.859823	0.906132	0.859823	0.986173	0.952440	0.5
5	CNNeq	flooding	S2	0.959354	0.947234	0.953255	0.947234	0.985383	0.975198	0.5
6	CNNeq	plateau	B	0.994723	0.998385	0.996551	0.998385	0.998058	0.998146	0.5
7	CNNeq	plateau	S1	0.957357	0.926624	0.941740	0.926624	0.984868	0.969243	0.5
8	CNNeq	plateau	S2	0.912784	0.927464	0.920065	0.927464	0.967510	0.956767	0.5
9	CNNeq	playback	B	0.991848	0.998922	0.995373	0.998922	0.997106	0.997579	0.5
10	CNNeq	playback	S1	0.958739	0.946098	0.952377	0.946098	0.985647	0.975339	0.5
11	CNNeq	playback	S2	0.922375	0.939050	0.930638	0.939050	0.972142	0.963517	0.5
12	CNNeq	suppress	B	0.995467	0.996775	0.996121	0.996775	0.998271	0.997858	0.5
13	CNNeq	suppress	S1	0.965347	0.803571	0.877061	0.803571	0.989010	0.937853	0.5
14	CNNeq	suppress	S2	0.963394	0.840898	0.897988	0.840898	0.987827	0.947294	0.5

In the tables 6.2 and 6.3, we investigate the predicted errors. All the predictions are in Table Predict. In the Error table, we can track the error and confirm which probe made the correct prediction. Using these two tables, we want to verify when an error in the prediction occurs, is it due to D_B or D_1 and D_2. In other words, can D_B give a good prediction in the case where D_1 and D_2 yield wrong predictions and vice-versa. Since our label setting is $y_1 = y_2 = y_B$, only case [0,0,0] or [1,1,1] are possible. The Table Predict 6.2 shows the different predictions obtained at each model on the test data. We define the Error Table 6.3 where the value 0 indicate that the prediction was correct by the concerning probe, and 1 means that the concerning probe commits an error. For example, the first row in the error table indicates that 55080 test examples have been correctly predicted. The colour red refers to the case [0,0,0], meaning a system is under attack, and all probes predict an attack. And blue refers to the case [1,1,1], meaning that the traffic is normal and all probes predicts normal traffic. We observe in the error table in indexes 1, 3,5 and 7 the different scenarios where D_B commit an error in prediction. In the index 1, both D_1 and D_2 returns a correct prediction while D_B return a false prediction. In that case, we also notice that most errors are blue (indicating False alarms).In indexes 3,5 and 7, D_B, S_1 and/or S_2 predict errors, unlike in indexes 2,4 and 6, where we notice that D_B returned the correct prediction even if D_1 and/or D_2 returned wrong predictions. Mainly from the error table, we observe more cases where the model D_B enhance the rendering of D_1 and D_2 for attack detection.

Table 6.2: Prediction table

	$\hat{y_1}$	$\hat{y_2}$	$\hat{y_B}$	
0	0	0	0	12922
1	0	0	1	42
2	0	1	0	1343
3	0	1	1	608
4	1	0	0	1667
5	1	0	1	1070
6	1	1	0	837
7	1	1	1	42206

Predict

Table 6.3: Error table

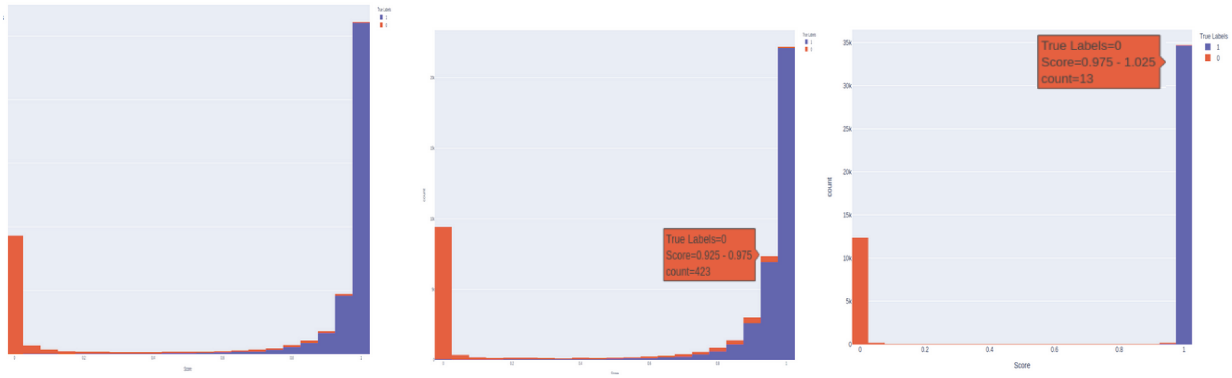
	E_1	E_2	E_b	
0	0	0	0	55080 (42161, 12919)
1	0	0	1	70 (68, 2)
2	0	1	0	2386(1055, 1331)
3	0	1	1	18(11, 7)
4	1	0	0	2257(601, 1656)
5	1	0	1	27(12, 15)
6	1	1	0	809(40, 769)
7	1	1	1	48(3, 45)

Error table

6.4.3 Communication overhead optimisation and metrics

In Fig 6.6 we observe the histogram distribution f the probabilities yield by each probe. We clearly see that the probability score is well balanced between anomalies and normal examples in D_B compared to D_1 and D_2, and it join the results obtained above Table 6.1. Based on D_i values, the goal is to define e_i, so we keep the result on D_B

performance while reducing the amount of communication between S_i and B. To this end, we need to set thresholds t_1 and t_2 based on the Eq 6.1. y_i defines how much the probes are certain of a given sequence's local behaviour normality. For this purpose, we need to obtain optimal values (t_1 and t_2) that maximise preventing communication between the probes (S_1, S_2) and S_B when the system behaviour is normal, and at the same time not impacting the performance on y_B. An exhaustive search is performed on threshold space (t_1, t_2) to find the tuple that maximises both communication optimisation without impacting the final result performance in terms of attack detection.



(a) S1 (b) S2 (c) B

Figure 6.6: Histograms of the distribution of probabilities score returned by D_1, D_2, D_B

In Table 6.4 we show the results obtained with a standard threshold selection at $t_i = 0.5$. Table 6.5 show the results obtained after gridsearch on the tuple values (t_1, t_2).

Table 6.4: Final Performance on DAD model on y_d. We evaluate the model on different attack classification. In this table the the threshold $t_1 = t_2 = 0.5$

atk	Precision	Recall	F1	Accuracy	opt_nor	opt_atk	error	good	Ne	gain
continuous	0.990212	0.979313	0.984733	0.991855	0.655946	0.265264	0.005088	0.073702	0.338966	0.661034
flooding	0.993971	0.988770	0.991364	0.995401	0.661409	0.265582	0.002589	0.070420	0.336002	0.663998
plateau	0.990302	0.976166	0.983183	0.991041	0.619061	0.264443	0.006273	0.110224	0.374666	0.625334
playback	0.989623	0.980595	0.985088	0.992262	0.654859	0.258256	0.004993	0.081892	0.340149	0.659851
suppress	0.990709	0.974319	0.982446	0.990395	0.570904	0.271307	0.006607	0.151182	0.422489	0.577511

In the table 6.4 we observe more gain compared to table 6.5. Indeed, the rate of communication optimisation (gain) can be more important if we neglect the error side effect, which directly impacts the DAD model's overall performance. In contrast, we observe that in table 6.5 the F1 measure is better than in the table 6.4. The tradeoff between optimisation and the model's performance in detecting an attack remains clear in our case, and we cannot tolerate a reduction of the model performance for the computational reason. So the table 6.5 show the final tunning of DAD model. Where we keep the same performance as before introducing the optimisation flag e_i. In Figs 6.7, 6.8, 6.9 we show the relation between the values of the tuple (t_1, t_2) and Ne (the rate of message sent from probes

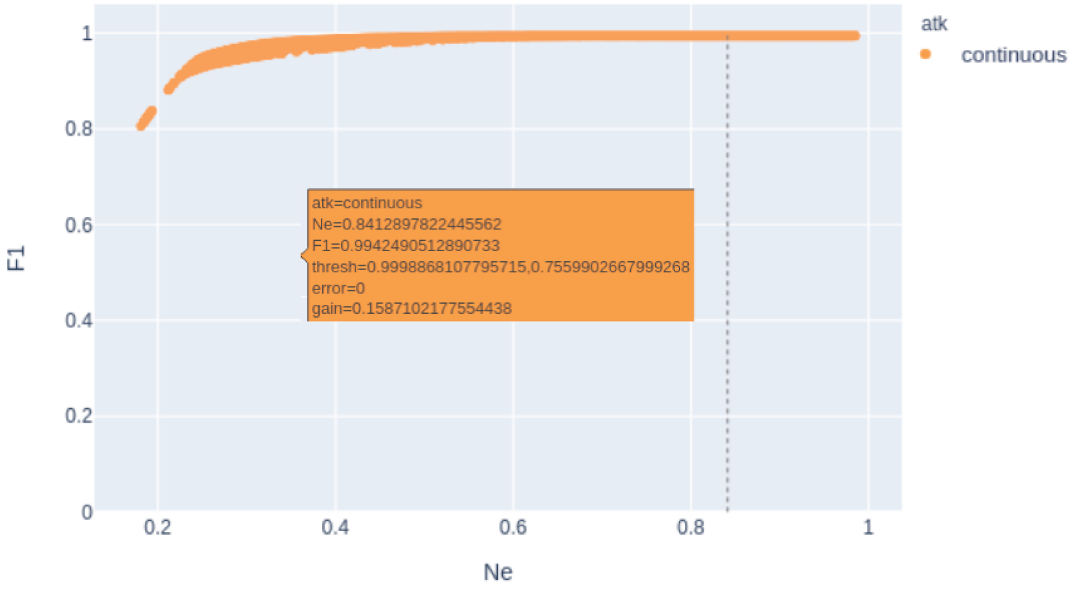
Table 6.5: Final Performance on DAD model on y_d. We evaluate the model on different attack classification. In this table the the threshold are found throw a gridsearch on threshold tuple (t_1, t_2) specific for each attack test data

atk	Precision	Recall	F1	Accuracy	opt_nor	opt_atk	error	good	Ne	gain
continuous	0.996549	0.991959	0.994249	0.996922	0.101863	0.266981	0.0	0.631156	0.898137	0.101863
flooding	0.994844	0.997256	0.996049	0.997888	0.033966	0.267626	0.0	0.698407	0.966034	0.033966
plateau	0.990486	0.988503	0.989493	0.994368	0.085081	0.267735	0.0	0.647184	0.914919	0.085081
playback	0.996078	0.989883	0.992971	0.996347	0.071691	0.259013	0.0	0.669296	0.928309	0.071691
suppress	0.997946	0.986503	0.992191	0.995716	0.028174	0.272708	0.0	0.699119	0.971826	0.028174

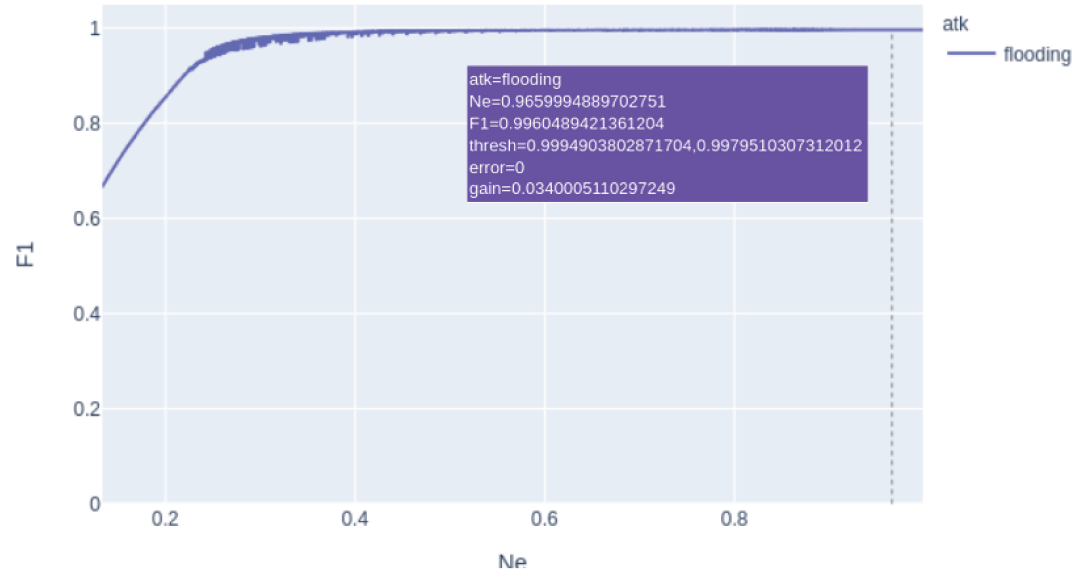
(S_1, S_2) to bastion B) and F1 measure metrics. The gridsearch on threshold space aims to find the tuple (t_1, t_2) that satisfy the strict condition of $error = 0$.

6.5 Conclusion

In-vehicle systems are getting more and more evolvement. This opens a new kind of architectures into the in-vehicle systems. Building an intrusion detection system needs to adapt to those architectures and their constraints. In this work, we formulated the integration of an IDS into future in-vehicle distributed architectures. We propose a framework DAD that fit the in-vehicle distributed architecture. DAD can capture local view patterns using the Temporal Convolutional Network (1DTCN)and send a reduced size feature vector to the system upper layer (bastion). Our framework also reduces the communication overhead brought by the intrusion detection system. We conduct a detailed experiment to assess our framework different prediction and analyse the various relations between the different probes prediction. We introduce new scores that take into consideration the optimisation metrics. Nonetheless, in this work, we set the e_i by manually searching threshold analysis through the t_i space. An improvement to the framework is introducing multiple-stage learning, where the communication overhead optimisation will be the second learning stage of the model. Future autonomous vehicles will use different protocols besides CAN bus, and other types of data can be aggregated from various sources. A dataset can lead the research community toward those problems because it is hard to simulate data that reflect the problem's real complexity.

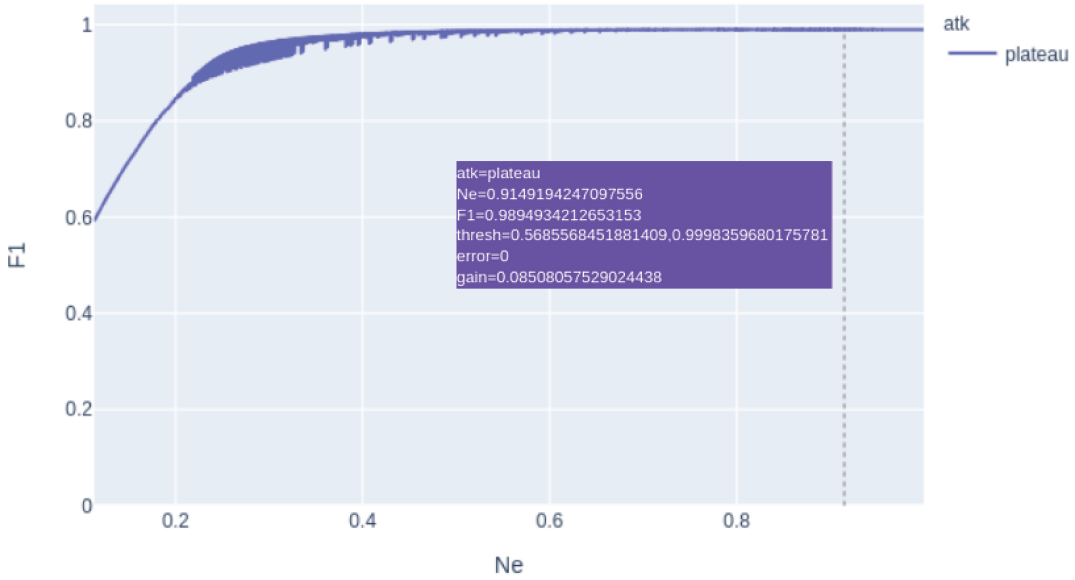


(a) Continues attack

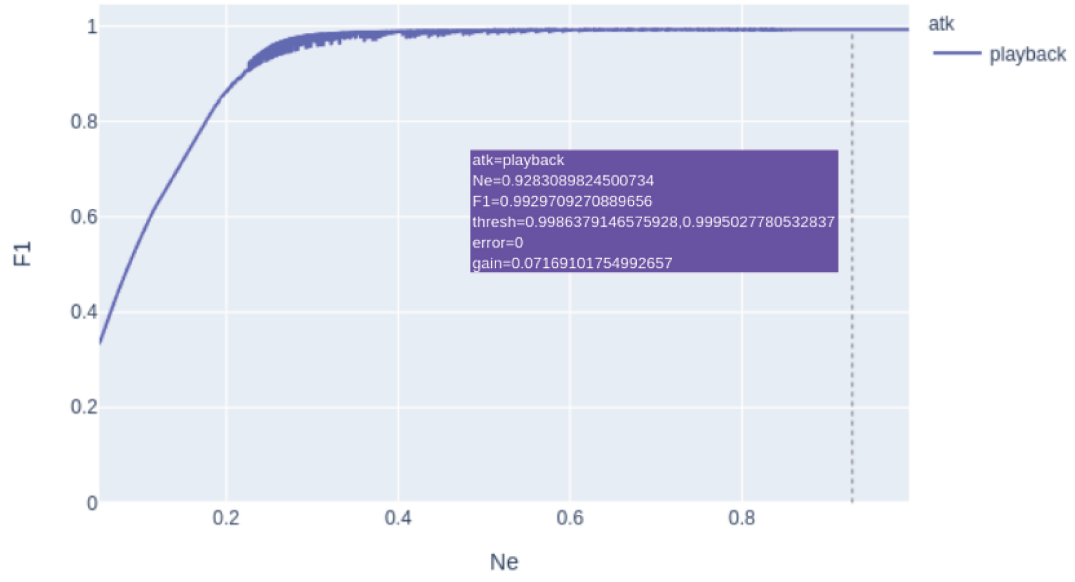


(b) Flooding attack

Figure 6.7: Visualisation of the F1 score and Ne based on different values of thresholds (t_1, t_2).

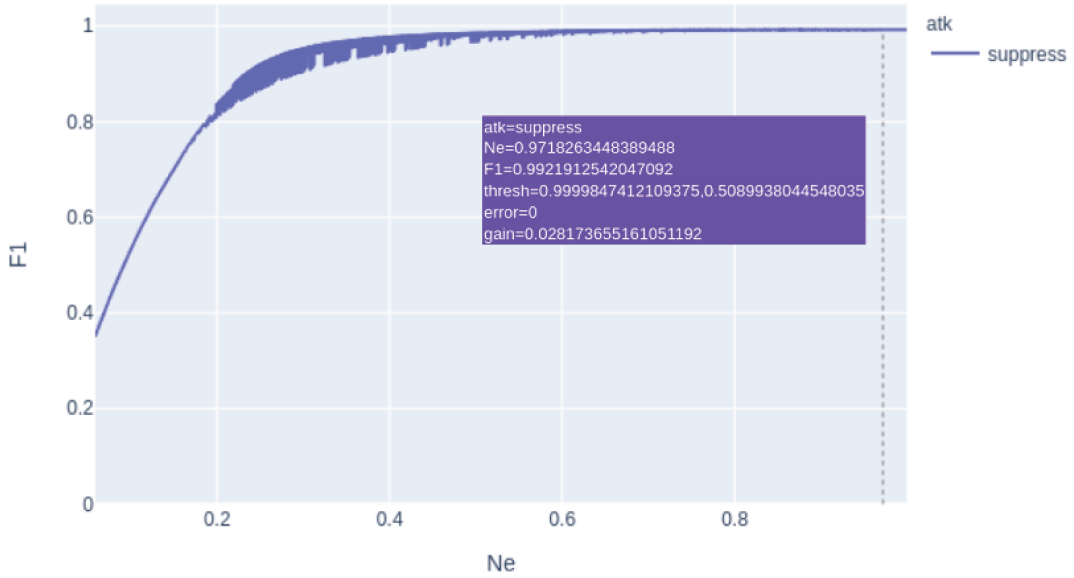


(a) Plateau attack



(b) Playback attack

Figure 6.8: Visualisation of the F1 score and Ne based on different values of thresholds (t_1, t_2).



(a) Suppress attack

Figure 6.9: Visualisation of the F1 score and Ne based on different values of thresholds (t_1, t_2).

Chapter 7

Conclusion

This thesis studies the use of Deep learning for in-vehicle Intrusion Detection System. This subject encompasses many fields, advanced fields like Deep learning for Anomaly Detection and Intrusion Detection systems, both applied in the relatively closed system (automotive industry) and relatively new, autonomous driving car as a context of the application. Our study reviewed the main methods used in the literature and the applied context's relevant characteristics (In-vehicle network architecture). We investigate Deep Learning as a solution to build an Intrusion Detection System. This first leads us to Generative Models, more specifically Adversarial Learning. Our first contribution, Chapter 4 is more general to the domain of Anomaly Detection and adversarial learning methods. Nonetheless, we already expect the need and constraints in memory in this approach that guided our intuitions and sharpened our results toward a lightweight solution. A logic continuation implies the analysis of the context (In-vehicle network). To this end, we go through the data which represent the system behaviour and topology. We first notice that compared to other application domains, the in-vehicle network data are not available. There is a lack of approved public dataset. This makes the different contributions of the state of the art of in-vehicle intrusion detection system diverge. The input also differs in terms of the end problem since it consists of solving the same problem using different inputs leading to different models and results that are not comparable. Our contribution Chapter 5 toward this significant issue is to propose a matrix representation for CAN data to improve the detection of different types of attack that impact both the content and the in-vehicle network flow. It also compares the different Deep Sequence modelling architecture to capture the normal behaviours pattern and detect various types of attacks. In the third part of this thesis, Chapter 6, we project further the in-vehicle network context in terms of evolvement to be near the actual prototypes network architecture of autonomous vehicles. The main question is how to fit the IDS using deep learning in this distributed and low resources environment. We proposed a framework (DAD) that matches the distributed and multi-view aspect of future in-vehicle networks. We proposed a framework that reduces the communication overhead without impacting the model's performance in detecting attacks.

7.1 Future work and improvements

This thesis contributes to different dimensions, theoretical and application domains. The general goal is to work on a framework that addresses some of the research questions related to building a deep learning-based Intrusion Detection System, which are not fully addressed by the scientific community. Multiple directions seem to be worth exploring, first as improving the actual contributions. Second, other research questions can be developed as a complementary contribution to developing a more general and complete In-vehicle Intrusion Detection System in terms of an end-to-end solution.

7.1.1 Contribution improvements

In the chapter 4 we introduced a new adversarial method called AnoEAn. The adversarial network is a robust framework for deep learning, enabling an implicit estimation of the data distribution p_X. In this work, we construct a decision space using a small amount of labelled anomalous examples. We notice that without introducing anomalous examples into the adversarial training process, the encoder tend to project any given data input into the chosen Gaussian distribution P_z. An improvement of this contribution consists of developing a one-class EAN. Recently, Chatillon und Ballester (2021) a research track grounded on the learning of the probability distribution P_x using a GAN learning strategy while simultaneously keeping track of the states of the associated generator discriminator during training. Secondly, they create a probability distribution (denoted as P_Ghist) that combines different states of the previous generator's history. An inspiration from this track applied to EAN is to use the probability distribution (P_Ehist) as an anomalous example.

A current straightforward work objective is to groups the different contribution in one general framework since the different contributions are independent in terms of problem modelling. At the same time, they all contribute to develop an in-vehicle intrusion detection system. The next step is to align the different contribution in one pipeline. The global framework is a distributed AnoEAN that inputs the occurrence matrix representation into a TCN Encoder for each view.

7.1.2 Interpretable Deep Learning for forensic analysis

In this work, we focus only on the detection step of an Intrusion Detection System. In the security domain, the forensic analysis of an attack after identifying the incident is critical to define the proper action that an autonomous system needs to take or a remote operator. Deep neural networks have been well-known for their superb performance, and in this thesis, we intensely studied their application into anomaly detection. However, due to their over-parameterized black-box nature, it is often difficult to understand the prediction results of deep models. In recent years, many interpretation tools have been proposed to explain or reveal the ways that deep models make

decisions Zhang u. a. (2020, 2021). Also, using attention guided anomaly detection Venkataramanan u. a. (2019) to augment the actual IDS feedback detection with information that can lead to an attack's source, aim, and location. Interpretability is believed to offer a sense of security by involving human in the decision-making process.

7.1.3 Adversarial examples

Due to its data-driven nature, Deep learning prediction is potentially susceptible to malicious manipulations to make errors on data that are surprisingly similar to examples the learned system handles correctly (Adversarial examples). The existence of these errors raises a variety of questions about out-of-sample generalization and whether bad actors might use such examples to abuse deployed systems. As a result of these security concerns, recent papers have flurry proposing algorithms to defend against such malicious perturbations of correctly handled examples Gilmer u. a. (2018); Goodfellow (2018); Liu u. a. (2018). It's essential to consider the adversarial examples techniques to make Deep Learning based Intrusion detection system more robust to those smart attacks and avoid introducing ever-increasing security concerns for those intelligent systems.

7.1.4 Common issues

In this work, we studied the applications of Deep learning algorithms, and we show how their usage is beneficial for the in-vehicle network. This study also highlights several shortcomings, such as the lack of datasets, the inability to learn from small datasets. Besides this, in-vehicle network data's complexity translates the complexity of building an appropriate IDS since Deep Learning are mainly data-driven approaches. A practical In-vehicle intrusion detection system needs to cope with the evolving aspect of the vehicle environment. To this end, we must construct the model and derive decision patterns from stream data produced by dynamically changing environments. The methods also need to develop an online sequential data processing to learn the nature of local subsystems and their interactions to endure self-organization of the system structure and parameters through time and contexts. Having data that express all those needs is nearly impossible. Some constructors already started collecting data on their different vehicles to improve their driving experience. But security-related data are heavy and take a lot of labour to extract and label since we need real attacks. Thus, it's important to think about an intrusion detection system's ability to develop and update itself to unknown environments and detect potential temporal shifts and drifts in input data.

Bibliography

[AIseg] AI literacy fundamentals. – Forschungsbericht

[oraclAIcs] ORACLE AND KPMG CLOUD THREAT REPORT / ORACLE AND KPMG. – Forschungsbericht. Accessed: 2019

[Abadi u. a. 2015] ABADI, Martín ; AGARWAL, Ashish ; BARHAM, Paul ; BREVDO, Eugene ; CHEN, Zhifeng ; CITRO, Craig ; CORRADO, Greg S. ; DAVIS, Andy ; DEAN, Jeffrey ; DEVIN, Matthieu ; GHEMAWAT, Sanjay ; GOODFELLOW, Ian ; HARP, Andrew ; IRVING, Geoffrey ; ISARD, Michael ; JIA, Yangqing ; JOZEFOWICZ, Rafal ; KAISER, Lukasz ; KUDLUR, Manjunath ; LEVENBERG, Josh ; MANÉ, Dandelion ; MONGA, Rajat ; MOORE, Sherry ; MURRAY, Derek ; OLAH, Chris ; SCHUSTER, Mike ; SHLENS, Jonathon ; STEINER, Benoit ; SUTSKEVER, Ilya ; TALWAR, Kunal ; TUCKER, Paul ; VANHOUCKE, Vincent ; VASUDEVAN, Vijay ; VIÉGAS, Fernanda ; VINYALS, Oriol ; WARDEN, Pete ; WATTENBERG, Martin ; WICKE, Martin ; YU, Yuan ; ZHENG, Xiaoqiang: *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. – URL https://www.tensorflow.org/. – Software available from tensorflow.org

[ABRAHAM und BOX 1979] ABRAHAM, BOVAS ; BOX, GEORGE E. P.: Bayesian analysis of some outlier problems in time series. In: *Biometrika* 66 (1979), 08, Nr. 2, S. 229–236. – URL https://doi.org/10.1093/biomet/66.2.229. – ISSN 0006-3444

[Afsin u. a. 2017] AFSIN, Mehmet E. ; SCHMIDT, Klaus W. ; SCHMIDT, Ece G.: C3:configurable CAN FD controller: architecture, design and hardware implementation. In: *2017 12th IEEE International Symposium on Industrial Embedded Systems (SIES)* (2017), S. 1–9

[Agarwal 2006] AGARWAL, Deepak: Detecting Anomalies in Cross-Classified Streams: A Bayesian Approach. In: *Knowl. Inf. Syst.* 11 (2006), Dezember, Nr. 1, S. 29–44. – ISSN 0219-1377

[Aggarwal 2013] AGGARWAL, Charu C.: *Outlier Analysis*. Springer, 2013. – URL http://dx.doi.org/10.1007/978-1-4614-6396-2. – ISBN 978-1-4614-6396-2

[Ahmed und Mahmood 2013] AHMED, M. ; MAHMOOD, A. N.: A novel approach for outlier detection and clustering improvement. In: *2013 IEEE 8th Conference on Industrial Electronics and Applications (ICIEA)*, 2013, S. 577–582

[Ahmed und Mahmood 2014] AHMED, M. ; MAHMOOD, A. N.: Network traffic analysis based on collective anomaly detection. In: *2014 9th IEEE Conference on Industrial Electronics and Applications*, 2014, S. 1141–1146

[Ahmed u. a. 2016a] AHMED, Mohiuddin ; MAHMOOD, Abdun N. ; ISLAM, Md. R.: A survey of anomaly detection techniques in financial domain. In: *Future Generation Computer Systems* 55 (2016), S. 278–288. – URL https://www.sciencedirect.com/science/article/pii/S0167739X15000023. – ISSN 0167-739X

[Ahmed u. a. 2016b] AHMED, Mohiuddin ; NASER MAHMOOD, Abdun ; HU, Jiankun: A Survey of Network Anomaly Detection Techniques. In: *J. Netw. Comput. Appl.* 60 (2016), Januar, Nr. C, S. 19–31. – URL https://doi.org/10.1016/j.jnca.2015.11.016. – ISSN 1084-8045

[Akbani u. a. 2004] AKBANI, Rehan ; KWEK, Stephen ; JAPKOWICZ, Nathalie: Applying Support Vector Machines to Imbalanced Datasets. In: *Proc. ECML 2004*, 2004, S. 39–50

[Akcay u. a. 2019] AKCAY, Samet ; ATAPOUR-ABARGHOUEI, Amir ; BRECKON, Toby P.: GANomaly: Semi-supervised Anomaly Detection via Adversarial Training. In: JAWAHAR, C. V. (Hrsg.) ; LI, Hongdong (Hrsg.) ; MORI, Greg (Hrsg.) ; SCHINDLER, Konrad (Hrsg.): *Computer Vision – ACCV 2018*. Cham : Springer International Publishing, 2019, S. 622–637

[Alcaraz u. a. 2017] ALCARAZ, C. ; LOPEZ, J. ; WOLTHUSEN, S.: OCPP Protocol: Security Threats and Challenges. In: *IEEE Transactions on Smart Grid* 8 (2017), Nr. 5, S. 2452–2459

[Alshammari u. a. 2018] ALSHAMMARI, Abdulaziz ; ZOHDY, Mohamed ; DEBNATH, Debatosh ; CORSER, George: Classification Approach for Intrusion Detection in Vehicle Systems. In: *Wireless Engineering and Technology* 09 (2018), 01, S. 79–94

[Anu und Vimala 2017] ANU, P. ; VIMALA, S.: A survey on sniffing attacks on computer networks. In: *2017 International Conference on Intelligent Computing and Control (I2C2)*, 2017, S. 1–5

[Asaka u. a. 1999] ASAKA, Midori ; TAGUCHI, Atsushi ; GOTO, Shigeki: *The Implementation of IDA: An Intrusion Detection Agent System*. 1999

[Avatefipour und Malik 2017] AVATEFIPOUR, Omid ; MALIK, Hafiz: State-of-the-Art Survey on In-Vehicle Network Communication "CAN-Bus" Security and Vulnerabilities. In: *International Journal of Computer Science and Network* 6 (2017), 12, S. 720–727

[Azmandian u. a. 2012] AZMANDIAN, F. ; YILMAZER, A. ; DY, J. G. ; ASLAM, J. A. ; KAELI, D. R.: GPU-Accelerated Feature Selection for Outlier Detection Using the Local Kernel Density Ratio. In: *2012 IEEE 12th International Conference on Data Mining*, 2012, S. 51–60

[Bagnall u. a. 2017] BAGNALL, Anthony ; LINES, Jason ; BOSTROM, Aaron ; LARGE, James ; KEOGH, Eamonn: The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. In: *Data Mining and Knowledge Discovery* 31 (2017), 05

[Bai u. a. 2018] BAI, Shaojie ; KOLTER, J. Z. ; KOLTUN, Vladlen: An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. In: *CoRR* abs/1803.01271 (2018)

[Baldi 2011] BALDI, Pierre: Autoencoders, Unsupervised Learning and Deep Architectures. In: *Proceedings of the 2011 International Conference on Unsupervised and Transfer Learning Workshop - Volume 27*, JMLR.org, 2011 (UTLW'11), S. 37–50

[Bariah u. a. 2015] BARIAH, L. ; SHEHADA, D. ; SALAHAT, E. ; YEUN, C. Y.: Recent Advances in VANET Security: A Survey. In: *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*, 2015, S. 1–7

[Barletta u. a. 2020] BARLETTA, Vita S. ; CAIVANO, Danilo ; NANNAVECCHIA, Antonella ; SCALERA, Michele: Intrusion Detection for in-Vehicle Communication Networks: An Unsupervised Kohonen SOM Approach. In: *Future Internet* 12 (2020), Nr. 7. – URL https://www.mdpi.com/1999-5903/12/7/119. – ISSN 1999-5903

[Basora u. a. 2019] BASORA, Luis ; OLIVE, Xavier ; DUBOT, Thomas: Recent Advances in Anomaly Detection Methods Applied to Aviation. In: *Aerospace* 6 (2019), Nr. 11. – URL https://www.mdpi.com/2226-4310/6/11/117. – ISSN 2226-4310

[Bayes 1763] BAYES, T.: An essay towards solving a problem in the doctrine of chances. In: *Phil. Trans. of the Royal Soc. of London* 53 (1763), S. 370–418

[Bellovin 2001] BELLOVIN, Steven M.: Computer Security—an End State? In: *Commun. ACM* 44 (2001), März, Nr. 3, S. 131–132. – URL https://doi.org/10.1145/365181.365241. – ISSN 0001-0782

[Bengio u. a. 2013] BENGIO, Y. ; COURVILLE, A. ; VINCENT, P.: Representation Learning: A Review and New Perspectives. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 35 (2013), Nr. 8, S. 1798–1828

[Berman u. a. 2019] BERMAN, Daniel S. ; BUCZAK, Anna L. ; CHAVIS, Jeffrey S. ; CORBETT, Cherita L.: A Survey of Deep Learning Methods for Cyber Security. In: *Information* 10 (2019), Nr. 4. – URL https://www.mdpi.com/2078-2489/10/4/122. – ISSN 2078-2489

[Bernardini u. a. 2017] BERNARDINI, Cesar ; ASGHAR, Muhammad R. ; CRISPO, Bruno: Security and privacy in vehicular communications: Challenges and opportunities. In: *Vehicular Communications* 10 (2017), S. 13 – 28. – URL http://www.sciencedirect.com/science/article/pii/S2214209617300803. – ISSN 2214-2096

[Bianco u. a. 2001] BIANCO, A. M. ; GARCÍA BEN, M. ; MARTÍNEZ, E. J. ; YOHAI, V. J.: Outlier Detection in Regression Models with ARIMA Errors using Robust Estimates. In: *Journal of Forecasting* 20 (2001), Nr. 8, S. 565–579. – URL https://onlinelibrary.wiley.com/doi/abs/10.1002/for.768

[Bijone 2016a] BIJONE, Manu: A Survey on Secure Network: Intrusion Detection & Prevention Approaches. In: *American Journal of Information Systems* 4 (2016), Nr. 3, S. 69–88. – URL http://pubs.sciepub.com/ajis/4/3/2. – ISSN 2374-1988

[Bijone 2016b] BIJONE, Manu: A Survey on Secure Network: Intrusion Detection & Prevention Approaches. In: *American Journal of Information Systems* 4 (2016), Nr. 3, S. 69–88. – URL http://pubs.sciepub.com/ajis/4/3/2. – ISSN 2374-1988

[Bishop 2006] BISHOP, Christopher M.: *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg : Springer-Verlag, 2006. – ISBN 0387310738

[Bostani und Sheikhan 2017] BOSTANI, Hamid ; SHEIKHAN, Mansour: Hybrid of anomaly-based and specification-based IDS for Internet of Things using unsupervised OPF based on MapReduce approach. In: *Computer Communications* 98 (2017), S. 52 – 71. – URL http://www.sciencedirect.com/science/article/pii/S0140366416306387. – ISSN 0140-3664

[Bostrom 2014] BOSTROM, Nick: *Superintelligence: Paths, Dangers, Strategies*. 1st. USA : Oxford University Press, Inc., 2014. – ISBN 0199678111

[Brauckhoff u. a. 2009] BRAUCKHOFF, Daniela ; DIMITROPOULOS, Xenofontas ; WAGNER, Arno ; SALAMATIAN, Kave: Anomaly Extraction in Backbone Networks Using Association Rules. In: *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference* 20 (2009), 11

[Brock u. a. 2018] BROCK, Andrew ; DONAHUE, Jeff ; SIMONYAN, Karen: Large Scale GAN Training for High Fidelity Natural Image Synthesis. In: *CoRR* abs/1809.11096 (2018). – URL http://arxiv.org/abs/1809.11096

[Brooks 1990] BROOKS, Rodney A.: Elephants don't play chess. In: *Robotics and Autonomous Systems* 6 (1990), Nr. 1, S. 3 – 15. – URL http://www.sciencedirect.com/science/article/pii/S0921889005800259. – Designing Autonomous Agents. – ISSN 0921-8890

[Brunner u. a. 2017] BRUNNER, Stefan ; RODER, Jurgen ; KUCERA, Markus ; WAAS, T.: Automotive E/E-architecture enhancements by usage of ethernet TSN. In: *2017 13th Workshop on Intelligent Solutions in Embedded Systems (WISES)* (2017), S. 9–13

[Buczak und Guven 2016] BUCZAK, A. L. ; GUVEN, E.: A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. In: *IEEE Communications Surveys Tutorials* 18 (2016), Nr. 2, S. 1153–1176

[Bulusu u. a. 2020] BULUSU, Saikiran ; KAILKHURA, B. ; LI, Bo ; VARSHNEY, P. ; SONG, D.: Anomalous Instance Detection in Deep Learning: A Survey. In: *ArXiv* abs/2003.06979 (2020)

[B.Zhao u. a. 2017] B.ZHAO ; H.LU ; S.CHEN ; J.LIU ; D.WU: Convolutional neural networks for time series classification. In: *Journal of Systems Engineering and Electronics* 28 (2017), Nr. 1, S. 162–169

[Cai u. a. 2019] CAI, Z. ; WANG, Aohui ; ZHANG, W.: - *1-0-days & Mitigations : Roadways to Exploit and Secure Connected BMW Cars*. 2019

[Campello u. a. 2015] CAMPELLO, Ricardo J. G. B. ; MOULAVI, Davoud ; ZIMEK, Arthur ; SANDER, Jörg: Hierarchical Density Estimates for Data Clustering, Visualization, and Outlier Detection. In: *TKDD* 10 (2015), Nr. 1, S. 5:1–5:51

[Candès u. a. 2011] CANDÈS, Emmanuel J. ; LI, Xiaodong ; MA, Yi ; WRIGHT, John: Robust Principal Component Analysis? In: *J. ACM* 58 (2011), Juni, Nr. 3. – URL https://doi.org/10.1145/1970392.1970395. – ISSN 0004-5411

[Carsten u. a. 2015] CARSTEN, Paul ; ANDEL, Todd R. ; YAMPOLSKIY, Mark ; MCDONALD, Jeffrey T.: In-Vehicle Networks: Attacks, Vulnerabilities, and Proposed Solutions. In: *Proceedings of the 10th Annual Cyber and Information Security Research Conference*. New York, NY, USA : Association for Computing Machinery, 2015 (CISR '15). – URL https://doi.org/10.1145/2746266.2746267. – ISBN 9781450333450

[Chalapathy und Chawla 2019] CHALAPATHY, Raghavendra ; CHAWLA, Sanjay: Deep Learning for Anomaly Detection: A Survey. In: *CoRR* abs/1901.03407 (2019). – URL http://arxiv.org/abs/1901.03407

[Chalapathy u. a. 2019] CHALAPATHY, Raghavendra ; TOTH, Edward ; CHAWLA, Sanjay: Group Anomaly Detection Using Deep Generative Models. In: BERLINGERIO, Michele (Hrsg.) ; BONCHI, Francesco (Hrsg.) ; GÄRTNER, Thomas (Hrsg.) ; HURLEY, Neil (Hrsg.) ; IFRIM, Georgiana (Hrsg.): *Machine Learning and Knowledge Discovery in Databases*. Cham : Springer International Publishing, 2019, S. 173–189

[Chandola u. a. 2009] CHANDOLA, Varun ; BANERJEE, Arindam ; KUMAR, Vipin: Anomaly Detection: A Survey. In: *ACM Computing Surveys* 41 (2009), Juli, Nr. 3. – URL https://doi.org/10.1145/1541880.1541882. – ISSN 0360-0300

[Chandola und Kumar 2007] CHANDOLA, Varun ; KUMAR, Vipin: Outlier Detection : A Survey. In: *ACM Computing Surveys* 41 (2007), 01

[Charlie Miller 2015] CHARLIE MILLER, Chris V.: *Remote Exploitation of an Unaltered Passenger Vehicle*. 2015

[Chatillon und Ballester 2021] CHATILLON, Pierrick ; BALLESTER, Coloma: History-Based Anomaly Detector: An Adversarial Approach to Anomaly Detection. In: ARAI, Kohei (Hrsg.) ; KAPOOR, Supriya (Hrsg.) ; BHATIA, Rahul

(Hrsg.): *Intelligent Systems and Applications*. Cham : Springer International Publishing, 2021, S. 761–776. – ISBN 978-3-030-55180-3

[Chattopadhyay u. a. 2020] CHATTOPADHYAY, A. ; LAM, K. ; TAVVA, Y.: Autonomous Vehicle: Security by Design. In: *IEEE Transactions on Intelligent Transportation Systems* (2020), S. 1–15

[Chawla 2003] CHAWLA, Nitesh V.: C4.5 and imbalanced data sets: investigating the effect of sampling method, probabilistic estimate, and decision tree structure. In: *In Proceedings of the ICML'03 Workshop on Class Imbalances*, 2003

[Checkoway u. a. 2011] CHECKOWAY, Stephen ; MCCOY, Damon ; KANTOR, Brian ; ANDERSON, Danny ; SHACHAM, Hovav ; SAVAGE, Stefan ; KOSCHER, Karl ; CZESKIS, Alexei ; ROESNER, Franziska ; KOHNO, Tadayoshi: Comprehensive Experimental Analyses of Automotive Attack Surfaces. In: *Proceedings of the 20th USENIX Conference on Security*. USA : USENIX Association, 2011 (SEC'11), S. 6

[CHEN u. a. 2005] CHEN, Da ; SHAO, Xueguang ; HU, Bin ; SU, Qingde: Simultaneous Wavelength Selection and Outlier Detection in Multivariate Regression of Near-Infrared Spectra. In: *Analytical Sciences* 21 (2005), Nr. 2, S. 161–166

[Cheswick u. a. 2003] CHESWICK, William R. ; BELLOVIN, Steven M. ; RUBIN, Aviel D.: *Firewalls and Internet Security: Repelling the Wily Hacker*. 2. USA : Addison-Wesley Longman Publishing Co., Inc., 2003. – ISBN 020163466X

[Cheung u. a. 2006] CHEUNG, S. ; DUTERTRE, B. ; FONG, M. ; LINDQVIST, U. ; SKINNER, K. ; VALDES, A.: Using Model-based Intrusion Detection for SCADA Networks, 2006

[Choi u. a. 2018] CHOI, W. ; JOO, K. ; JO, H. J. ; PARK, M. C. ; LEE, D. H.: VoltageIDS: Low-Level Communication Characteristics for Automotive Intrusion Detection System. In: *IEEE Transactions on Information Forensics and Security* 13 (2018), Nr. 8, S. 2114–2129

[Choi u. a. 2016] CHOI, Wonsuk ; JO, Hyo ; WOO, Samuel ; CHUN, Ji ; PARK, Jooyoung: Identifying ECUs Using Inimitable Characteristics of Signals in Controller Area Networks. In: *IEEE Transactions on Vehicular Technology* PP (2016), 07

[Chollet u. a. 2015] CHOLLET ; FRANCOIS u. a.: *Keras*. 2015

[claroty 2020] CLAROTY: *CLAROTY BIANNUAL ICS RISK and VULNERABILITY REPORT: 1H 2020*. 2020

[Corrigan 2002] CORRIGAN, Steve: Introduction to the Controller Area Network (CAN). (2002)

[Cortes und Vapnik 1995] CORTES, C. ; VAPNIK, V.: Support Vector Networks. In: *Machine Learning* 20 (1995), S. 273–297

[Coulouris u. a. 2011] COULOURIS, George ; DOLLIMORE, Jean ; KINDBERG, Tim ; BLAIR, Gordon: *Distributed Systems: Concepts and Design*. 5th. USA : Addison-Wesley Publishing Company, 2011. – ISBN 0132143011

[Creswell und Bharath 2019] CRESWELL, A. ; BHARATH, A. A.: Inverting the Generator of a Generative Adversarial Network. In: *IEEE Transactions on Neural Networks and Learning Systems* 30 (2019), Nr. 7, S. 1967–1974

[Creswell u. a. 2018] CRESWELL, Antonia ; WHITE, Tom ; DUMOULIN, Vincent ; ARULKUMARAN, Kai ; SENGUPTA, Biswa ; BHARATH, Anil A.: Generative Adversarial Networks: An Overview. In: *IEEE Signal Process. Mag.* 35 (2018), Nr. 1, S. 53–65

[CYPRESS 2018] CYPRESS: Automotive security / infineon. 2018. – Forschungsbericht

[Dara und Tumma 2018] DARA, S. ; TUMMA, P.: Feature Extraction By Using Deep Learning: A Survey. In: *2018 Second International Conference on Electronics, Communication and Aerospace Technology (ICECA)*, 2018, S. 1795–1801

[Debar u. a. 1999] DEBAR, Hervé ; DACIER, Marc ; WESPI, Andreas: Towards a taxonomy of intrusion-detection systems. In: *Computer Networks* 31 (1999), Nr. 8, S. 805 – 822. – URL http://www.sciencedirect.com/science/article/pii/S1389128698000176. – ISSN 1389-1286

[Deecke u. a. 2018] DEECKE, Lucas ; VANDERMEULEN, Robert A. ; RUFF, Lukas ; MANDT, Stephan ; KLOFT, Marius: Image Anomaly Detection with Generative Adversarial Networks. In: *Proc. ECML 2018*, 2018, S. 3–17

[Denning 1987] DENNING, D. E.: An Intrusion-Detection Model. In: *IEEE Transactions on Software Engineering* SE-13 (1987), Nr. 2, S. 222–232

[Dibaei u. a. 2019] DIBAEI, Mahdi ; ZHENG, X. ; JIANG, Kun ; MARIC, S. ; ABBAS, Robert ; LIU, S. ; ZHANG, Yuexin ; DENG, Yao ; WEN, Sheng ; ZHANG, Jun ; XIANG, Y. ; YU, Shui: An Overview of Attacks and Defences on Intelligent Connected Vehicles. In: *ArXiv* (2019)

[Ding u. a. 2019] DING, Kaize ; LI, Jundong ; BHANUSHALI, Rohit ; LIU, Huan: Deep Anomaly Detection on Attributed Networks. In: *Proceedings of the 2019 SIAM International Conference on Data Mining (SDM)*, URL https://epubs.siam.org/doi/abs/10.1137/1.9781611975673.67, 2019, S. 594–602

[Donahue u. a. 2017] DONAHUE, J. ; HENDRICKS, L. A. ; ROHRBACH, M. ; VENUGOPALAN, S. ; GUADARRAMA, S. ; SAENKO, K. ; DARRELL, T.: Long-Term Recurrent Convolutional Networks for Visual Recognition and Description. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39 (2017), Nr. 4, S. 677–691

[Donahue u. a. 2016] DONAHUE, Jeff ; KRÄHENBÜHL, Philipp ; DARRELL, Trevor: Adversarial Feature Learning. In: *CoRR* abs/1605.09782 (2016)

[Dorothy und Neumann 1985] DOROTHY, Denning ; NEUMANN, Peter G.: *Requirements and Model for IDES - a Real-time Intrusion-detection Expert System: Final Report*. SRI International, 1985

[Duan und Keerthi 2005] DUAN, Kai-Bo ; KEERTHI, S. S.: Which is the Best Multiclass SVM Method? An Empirical Study. In: *Proceedings of the 6th International Conference on Multiple Classifier Systems*. Berlin, Heidelberg : Springer-Verlag, 2005 (MCS'05), S. 278–285. – URL https://doi.org/10.1007/11494683_28. – ISBN 3540263063

[Dumoulin u. a. 2016] DUMOULIN, Vincent ; BELGHAZI, Ishmael ; POOLE, Ben ; LAMB, Alex ; ARJOVSKY, Martín ; MASTROPIETRO, Olivier ; COURVILLE, Aaron C.: Adversarially Learned Inference. In: *CoRR* abs/1606.00704 (2016)

[Dupont u. a. 2019] DUPONT, G. ; DEN HARTOG, J. ; ETALLE, S. ; LEKIDIS, A.: A survey of network intrusion detection systems for controller area network. In: *2019 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*, 2019, S. 1–6

[El-Rewini u. a. 2020] EL-REWINI, Zeinab ; SADATSHARAN, Karthikeyan ; SELVARAJ, Daisy F. ; PLATHOTTAM, Siby J. ; RANGANATHAN, Prakash: Cybersecurity challenges in vehicular communications. In: *Vehicular Communications* 23 (2020), S. 100214. – URL http://www.sciencedirect.com/science/article/pii/S221420961930261X. – ISSN 2214-2096

[Engen 2010] ENGEN, Vegard: *Machine learning for network based intrusion detection: an investigation into discrepancies in findings with the KDD cup '99 data set and multi-objective evolution of neural network classifier ensembles from imbalanced data*. June 2010. – URL http://eprints.bournemouth.ac.uk/15899/

[Engoulou u. a. 2014] ENGOULOU, Richard G. ; BELLAÏCHE, Martine ; PIERRE, Samuel ; QUINTERO, Alejandro: VANET security surveys. In: *Computer Communications* 44 (2014), S. 1 – 13. – URL http://www.sciencedirect.com/science/article/pii/S0140366414000863. – ISSN 0140-3664

[ENISA 2019] ENISA: *ENISA GOOD PRACTICES FOR SECURITY OF SMART CARS*. 2019

[Eskin 2000] ESKIN, Eleazar: Anomaly Detection over Noisy Data Using Learned Probability Distributions. In: *Proceedings of the Seventeenth International Conference on Machine Learning*. San Francisco, CA, USA : Morgan Kaufmann Publishers Inc., 2000 (ICML '00), S. 255–262. – ISBN 1558607072

[Eskin u. a. 2002] ESKIN, Eleazar ; ARNOLD, Andrew ; PRERAU, Michael ; PORTNOY, Leonid ; STOLFO, Sal: *A Geometric Framework for Unsupervised Anomaly Detection*. S. 77–101. In: BARBARÁ, Daniel (Hrsg.) ; JAJODIA, Sushil (Hrsg.): *Applications of Data Mining in Computer Security*. Boston, MA : Springer US, 2002. – URL https://doi.org/10.1007/978-1-4615-0953-0_4. – ISBN 978-1-4615-0953-0

[Fagnant und Kockelman 2015] FAGNANT, Daniel ; KOCKELMAN, Kara: Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations. In: *Transportation Research Part A: Policy and Practice* 77 (2015), 07

[Farsi u. a. 1999] FARSI, M. ; RATCLIFF, K. ; BARBOSA, M.: An overview of controller area network. In: *Computing Control Engineering Journal* 10 (1999), Nr. 3, S. 113–120

[Feng u. a. 2014] FENG, Fangxiang ; WANG, Xiaojie ; LI, Ruifan: Cross-Modal Retrieval with Correspondence Autoencoder. New York, NY, USA : Association for Computing Machinery, 2014 (MM '14), S. 7–16. – URL https://doi.org/10.1145/2647868.2654902. – ISBN 9781450330633

[Forest und Jochim 2011] FOREST, Thomas ; JOCHIM, Markus: On the Fault Detection Capabilities of AUTOSAR's End-to-End Communication Protection CRC's. In: *SAE Technical Paper*, SAE International, 04 2011. – URL https://doi.org/10.4271/2011-01-0999

[Francillon u. a. 2010] FRANCILLON, Aurélien ; DANEV, Boris ; CAPKUN, Srdjan: Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars. In: *IACR Cryptology ePrint Archive* 2010 (2010), 01, S. 332

[Fries und Falk 2012] FRIES, Steffen ; FALK, Rainer: Electric Vehicle Charging Infrastructure – Security Considerations and Approaches, 06 2012

[Fukushima 1980] FUKUSHIMA, Kunihiko: Neocognitron: A Self-Organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position. In: *Biological Cybernetics* 36 (1980), S. 193–202

[G. Leen u. a. 1999] G. LEEN ; HEFFERNAN, D. ; DUNNE, A.: Digital networks in the automotive vehicle. In: *Computing & Control Engineering Journal* 10 (1999), December, S. 257–266(9). – URL https://digital-library.theiet.org/content/journals/10.1049/cce_19990604. – ISSN 0956-3385

[Galinina u. a. 2018] GALININA, Olga (Hrsg.) ; ANDREEV, Sergey (Hrsg.) ; BALANDIN, Sergey I. (Hrsg.) ; KOUCHERYAVY, Yevgeni (Hrsg.): *State of the Art Literature Review on Network Anomaly Detection with Deep Learning*. Bd. *11118*. 2018. (Lecture Notes in Computer Science)

[García-Teodoro u. a. 2009] GARCÍA-TEODORO, P. ; DÍAZ-VERDEJO, J. ; MACIÁ-FERNÁNDEZ, G. ; VÁZQUEZ, E.: Anomaly-Based Network Intrusion Detection: Techniques, Systems and Challenges. In: *Comput. Secur.* 28 (2009), Februar, Nr. 1–2, S. 18–28. – URL https://doi.org/10.1016/j.cose.2008.08.003. – ISSN 0167-4048

[Garg und Maheshwari 2016] GARG, A. ; MAHESHWARI, P.: A hybrid intrusion detection system: A review. In: *2016 10th International Conference on Intelligent Systems and Control (ISCO)*, 2016, S. 1–5

[Gharibian und Ghorbani 2007] GHARIBIAN, F. ; GHORBANI, A. A.: Comparative Study of Supervised Machine Learning Techniques for Intrusion Detection. In: *Fifth Annual Conference on Communication Networks and Services Research (CNSR '07)*, 2007, S. 350–358

[Gilmer u. a. 2018] GILMER, J. ; ADAMS, R. ; GOODFELLOW, Ian J. ; ANDERSEN, David G. ; DAHL, George E.: Motivating the Rules of the Game for Adversarial Example Research. In: *ArXiv* abs/1807.06732 (2018)

[Golan und El-Yaniv 2018] GOLAN, Izhak ; EL-YANIV, Ran: Deep Anomaly Detection Using Geometric Transformations. In: *Proc. NIPS 2018*, 2018, S. 9781–9791

[Gonog und Zhou 2019] GONOG, L. ; ZHOU, Y.: A Review: Generative Adversarial Networks. In: *2019 14th IEEE Conference on Industrial Electronics and Applications (ICIEA)*, 2019, S. 505–510

[Good 1966] GOOD, Irving J.: Speculations Concerning the First Ultraintelligent Machine. Elsevier, 1966, S. 31 – 88. – URL http://www.sciencedirect.com/science/article/pii/S0065245808604180. – ISSN 0065-2458

[Goodfellow u. a. 2016] GOODFELLOW, Ian ; BENGIO, Yoshua ; COURVILLE, Aaron: *Deep Learning*. MIT Press, 2016

[Goodfellow 2017] GOODFELLOW, Ian J.: NIPS 2016 Tutorial: Generative Adversarial Networks. In: *CoRR* abs/1701.00160 (2017). – URL http://arxiv.org/abs/1701.00160

[Goodfellow 2018] GOODFELLOW, Ian J.: Defense Against the Dark Arts: An overview of adversarial example security research and future research directions. In: *CoRR* abs/1806.04169 (2018). – URL http://arxiv.org/abs/1806.04169

[Goodfellow u. a. 2014] GOODFELLOW, Ian J. ; POUGET-ABADIE, Jean ; MIRZA, Mehdi ; XU, Bing ; WARDE-FARLEY, David ; OZAIR, Sherjil ; COURVILLE, Aaron ; BENGIO, Yoshua: Generative Adversarial Nets. In: *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*. Cambridge, MA, USA : MIT Press, 2014 (NIPS'14), S. 2672–2680

[Govaert und Nadif 2014] GOVAERT, Gérard ; NADIF, Mohamed: *Co-Clustering*. ISTE-Wiley, 2014 (Computing Engineering series). – 256 S. – URL https://hal.archives-ouvertes.fr/hal-00933301

[Govaert und Nadif 2008] GOVAERT, Gérard ; NADIF, Mohamed: Block clustering with Bernoulli mixture models: Comparison of different approaches. In: *Computational Statistics & Data Analysis* 52 (2008), Nr. 6, S. 3233–3245. – URL https://www.sciencedirect.com/science/article/pii/S0167947307003441. – ISSN 0167-9473

[Groll und Ruland 2009] GROLL, A. ; RULAND, C.: Secure and authentic communication on existing in-vehicle networks. In: *2009 IEEE Intelligent Vehicles Symposium*, 2009, S. 1093–1097

[Groza und Murvay 2018] GROZA, B. ; MURVAY, P.: Security Solutions for the Controller Area Network: Bringing Authentication to In-Vehicle Networks. In: *IEEE Vehicular Technology Magazine* 13 (2018), Nr. 1, S. 40–47

[Guha und Kachirski 2003] GUHA, R. ; KACHIRSKI, O.: Effective Intrusion Detection Using Multiple Sensors in Wireless Ad Hoc Networks. In: *2014 47th Hawaii International Conference on System Sciences* Bd. 3. Los Alamitos, CA, USA : IEEE Computer Society, jan 2003, S. 57a. – URL https://doi.ieeecomputersociety.org/10.1109/HICSS.2003.1173873

[Haas und Langjahr 2016] HAAS, Waldemar ; LANGJAHR, P.: Cross-domain vehicle control units in modern E/E architectures. In: BARGENDE, Michael (Hrsg.) ; REUSS, Hans-Christian (Hrsg.) ; WIEDEMANN, Jochen (Hrsg.): *16. Internationales Stuttgarter Symposium*. Wiesbaden : Springer Fachmedien Wiesbaden, 2016, S. 1619–1627

[Hahn u. a. 2019] HAHN, D. A. ; MUNIR, A. ; BEHZADAN, V.: Security and Privacy Issues in Intelligent Transportation Systems: Classification and Challenges. In: *IEEE Intelligent Transportation Systems Magazine* (2019), S. 1–1

[Hahn u. a. 2019] HAHN, Dalton A. ; MUNIR, A. ; BEHZADAN, Vahid: Security and Privacy Issues in Intelligent Transportation Systems: Classification and Challenges. In: *IEEE Intelligent Transportation Systems Magazine* (2019), S. 1–1

[Hamada u. a. 2018] HAMADA, Yoshihiro ; INOUE, Masayuki ; UEDA, Hiroshi ; MIYASHITA, Yukihiro ; HATA, Yoichi: Anomaly-Based Intrusion Detection Using the Density Estimation of Reception Cycle Periods for In-Vehicle Networks. In: *SAE Int. J. Transp. Cyber. & Privacy* 1 (2018), 05, S. 39–56. – URL https://doi.org/10.4271/11-01-01-0003

[Hamed u. a. 2018] HAMED, Tarfa ; ERNST, Jason B. ; KREMER, Stefan C.: *A Survey and Taxonomy of Classifiers of Intrusion Detection Systems*. S. 21–39. In: *Computer and Network Security Essentials*. Cham : Springer International Publishing, 2018. – URL https://doi.org/10.1007/978-3-319-58424-9_2

[Han u. a. 2021] HAN, Z. ; ZHAO, J. ; LEUNG, H. ; MA, K. F. ; WANG, W.: A Review of Deep Learning Models for Time Series Prediction. In: *IEEE Sensors Journal* 21 (2021), Nr. 6, S. 7833–7848

[Hanselmann u. a. 2020] HANSELMANN, Markus ; STRAUSS, Thilo ; DORMANN, Katharina ; ULMER, Holger: CANet: An Unsupervised Intrusion Detection System for High Dimensional CAN Bus Data. In: *IEEE Access* 8 (2020), S. 58194–58205

[Hartwich und Bosch 2012] HARTWICH, F. ; BOSCH, R.: CAN with Flexible Data-Rate, 2012

[Hasrouny u. a. 2017] HASROUNY, Hamssa ; SAMHAT, Abed E. ; BASSIL, Carole ; LAOUITI, Anis: VANet security challenges and solutions: A survey. In: *Vehicular Communications* 7 (2017), S. 7 – 20. – URL http://www.sciencedirect.com/science/article/pii/S2214209616301231. – ISSN 2214-2096

[Hawkins 1980] HAWKINS, D. M.: *Identification of outliers*. London [u.a.] : Chapman and Hall, 1980 (Monographs on applied probability and statistics)

[Hawkins u. a. 2002] HAWKINS, Simon ; HE, Hongxing ; WILLIAMS, Graham J. ; BAXTER, Rohan A.: Outlier Detection Using Replicator Neural Networks. In: *Proceedings of the 4th International Conference on Data Warehousing and Knowledge Discovery*. Berlin, Heidelberg : Springer-Verlag, 2002 (DaWaK 2000), S. 170–180. – ISBN 3540441239

[Hindy u. a. 2020] HINDY, H. ; BROSSET, D. ; BAYNE, E. ; SEEAM, A. K. ; TACHTATZIS, C. ; ATKINSON, R. ; BELLEKENS, X.: A Taxonomy of Network Threats and the Effect of Current Datasets on Intrusion Detection Systems. In: *IEEE Access* 8 (2020), S. 104650–104675

[Hinton und Salakhutdinov 2006] HINTON, G. E. ; SALAKHUTDINOV, R. R.: Reducing the Dimensionality of Data with Neural Networks. In: *Science* 313 (2006), Nr. 5786, S. 504–507. – URL https://science.sciencemag.org/content/313/5786/504. – ISSN 0036-8075

[Hochreiter und Schmidhuber 1997] HOCHREITER, Sepp ; SCHMIDHUBER, Jürgen: Long Short-Term Memory. In: *Neural Comput.* 9 (1997), November, Nr. 8, S. 1735–1780. – ISSN 0899-7667

[Hodge und Austin 2004] HODGE, Victoria J. ; AUSTIN, Jim: A Survey of Outlier Detection Methodologies. In: *Artif. Intell. Rev.* 22 (2004), Nr. 2, S. 85–126

[Hodo u. a. 2017] HODO, Elike ; BELLEKENS, Xavier ; HAMILTON, Andrew ; TACHTATZIS, Christos ; ATKINSON, Robert: *Shallow and Deep Networks Intrusion Detection System: A Taxonomy and Survey*. 2017

[Hoppe u. a. 2011] HOPPE, Tobias ; KILTZ, Stefan ; DITTMANN, Jana: Security threats to automotive CAN networks—Practical examples and selected short-term countermeasures. In: *Reliability Engineering & System Safety* 96 (2011), Nr. 1, S. 11–25. – URL https://www.sciencedirect.com/science/article/pii/S0951832010001602. – Special Issue on Safecomp 2008. – ISSN 0951-8320

[Ismail Fawaz u. a. 2019] ISMAIL FAWAZ, Hassan ; FORESTIER, Germain ; WEBER, Jonathan ; IDOUMGHAR, Lhassane ; MULLER, Pierre-Alain: Deep learning for time series classification: a review. In: *Data Mining and Knowledge Discovery* 33 (2019), 07

[Jadoon u. a. 2018] JADOON, Ahmer ; WANG, Licheng ; LI, Tong ; ZIA, Muhammad: Lightweight Cryptographic Techniques for Automotive Cybersecurity. In: *Wireless Communications and Mobile Computing* 2018 (2018), 06, S. 1–15

[Jain u. a. 1999] JAIN, A. K. ; MURTY, M. N. ; FLYNN, P. J.: Data Clustering: A Review. In: *ACM Comput. Surv.* 31 (1999), September, Nr. 3, S. 264–323. – URL https://doi.org/10.1145/331499.331504. – ISSN 0360-0300

[Ji u. a. 2018] JI, H. ; WANG, Y. ; QIN, H. ; WANG, Y. ; LI, H.: Comparative Performance Evaluation of Intrusion Detection Methods for In-Vehicle Networks. In: *IEEE Access* 6 (2018), S. 37523–37532

[Ji u. a. 2019] JI, Y. ; HUANG, L. ; HE, H. ; WANG, C. ; XIE, G. ; SHI, W. ; LIN, K.: Multi-view Outlier Detection in Deep Intact Space. In: *2019 IEEE International Conference on Data Mining (ICDM)*, 2019, S. 1132–1137

[Jiang und Cybenko 2004] JIANG, G. ; CYBENKO, G.: Temporal and spatial distributed event correlation for network security. In: *Proceedings of the 2004 American Control Conference* Bd. 2, 2004, S. 996–1001 vol.2

[Jo u. a. 2017] JO, Hyo J. ; CHOI, Wonsuk ; NA, Seoung Y. ; WOO, Samuel ; LEE, Dong H.: Vulnerabilities of Android OS-Based Telematics System. In: *Wirel. Pers. Commun.* 92 (2017), Februar, Nr. 4, S. 1511–1530. – URL https://doi.org/10.1007/s11277-016-3618-9. – ISSN 0929-6212

[Kalkan und Sahingoz 2020] KALKAN, S. C. ; SAHINGOZ, O. K.: In-Vehicle Intrusion Detection System on Controller Area Network with Machine Learning Models. In: *2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 2020, S. 1–6

[Kang u. a. 2017] KANG, K. ; BAEK, Y. ; LEE, S. ; SON, S. H.: An Attack-Resilient Source Authentication Protocol in Controller Area Network. In: *2017 ACM/IEEE Symposium on Architectures for Networking and Communications Systems (ANCS)*, 2017, S. 109–118

[Kang und Kang 2016] KANG, M. ; KANG, J.: A Novel Intrusion Detection Method Using Deep Neural Network for In-Vehicle Network Security. In: *2016 IEEE 83rd Vehicular Technology Conference (VTC Spring)*, 2016, S. 1–5

[Kang und Kang 2016] KANG, Min-Joo ; KANG, Je-Won: Intrusion Detection System Using Deep Neural Network for In-Vehicle Network Security. In: *PLOS ONE* 11 (2016), 06, S. 1–17

[Kargl u. a. 2008] KARGL, F. ; PAPADIMITRATOS, P. ; BUTTYAN, L. ; MÜTER, M. ; SCHOCH, E. ; WIEDERSHEIM, B. ; THONG, T. ; CALANDRIELLO, G. ; HELD, A. ; KUNG, A. ; HUBAUX, J.: Secure vehicular communication systems: implementation, performance, and research challenges. In: *IEEE Communications Magazine* 46 (2008), Nr. 11, S. 110–118

[Karras u. a. 2017] KARRAS, Tero ; AILA, Timo ; LAINE, Samuli ; LEHTINEN, Jaakko: Progressive Growing of GANs for Improved Quality, Stability, and Variation. In: *CoRR* abs/1710.10196 (2017). – URL http://arxiv.org/abs/1710.10196

[Keller u. a. 2012] KELLER, F. ; MULLER, E. ; BOHM, K.: HiCS: High Contrast Subspaces for Density-Based Outlier Ranking. In: *2012 IEEE 28th International Conference on Data Engineering*, 2012, S. 1037–1048

[Kendall 1999] KENDALL, K.: A Database of Computer Attacks for the Evaluation of Intrusion Detection Systems, 1999

[Khan u. a. 2017] KHAN, Saad ; PARKINSON, Simon ; QIN, Yongrui: Fog Computing Security: A Review of Current Applications and Security Solutions. In: *J. Cloud Comput.* 6 (2017), Dezember, Nr. 1. – URL https://doi.org/10.1186/s13677-017-0090-3. – ISSN 2192-113X

[Kingma u. a. 2014] KINGMA ; DIEDERIK ; BA, Jimmy: Adam: A Method for Stochastic Optimization. In: *International Conference on Learning Representations* (2014), 12

[Kiran u. a. 2018] KIRAN, B. R. ; THOMAS, Dilip M. ; PARAKKAL, Ranjith: An Overview of Deep Learning Based Methods for Unsupervised and Semi-Supervised Anomaly Detection in Videos. In: *J. Imaging* 4 (2018), Nr. 2, S. 36

[Kiss u. a. 2014] KISS, I. ; GENGE, B. ; HALLER, P. ; SEBESTYÉN, G.: Data clustering-based anomaly detection in industrial control systems. In: *2014 IEEE 10th International Conference on Intelligent Computer Communication and Processing (ICCP)*, 2014, S. 275–281

[Ko u. a. 1994] KO, C. ; FINK, G. ; LEVITT, K.: Automated detection of vulnerabilities in privileged programs by execution monitoring. In: *Tenth Annual Computer Security Applications Conference*, 1994, S. 134–144

[Koscher u. a. 2010a] KOSCHER, K. ; CZESKIS, A. ; ROESNER, F. ; PATEL, S. ; KOHNO, T. ; CHECKOWAY, S. ; MCCOY, D. ; KANTOR, B. ; ANDERSON, D. ; SHACHAM, H. ; SAVAGE, S.: Experimental Security Analysis of a Modern Automobile. In: *2010 IEEE Symposium on Security and Privacy*, 2010, S. 447–462

[Koscher u. a. 2010b] KOSCHER, K. ; CZESKIS, A. ; ROESNER, F. ; PATEL, S. ; KOHNO, T. ; CHECKOWAY, S. ; MCCOY, D. ; KANTOR, B. ; ANDERSON, D. ; SHACHAM, H. ; SAVAGE, S.: Experimental Security Analysis of a Modern Automobile. In: *2010 IEEE Symposium on Security and Privacy*, 2010, S. 447–462

[Kozik und Choraś 2016] KOZIK, Rafał ; CHORAŚ, Michał: Solution to Data Imbalance Problem in Application Layer Anomaly Detection Systems. In: *Hybrid Artificial Intelligent Systems*. Cham : Springer International Publishing, 2016, S. 441–450. – ISBN 978-3-319-32034-2

[Kruegel 2004] KRUEGEL, Christopher: *Intrusion Detection and Correlation: Challenges and Solutions*. Santa Clara, CA, USA : Springer-Verlag TELOS, 2004. – ISBN 0387233989

[Kruegel und Toth 2003] KRUEGEL, Christopher ; TOTH, Thomas: Using Decision Trees to Improve Signature-Based Intrusion Detection. In: *Recent Advances in Intrusion Detection*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2003, S. 173–191

[Krügel u. a. 2002] KRÜGEL, Christopher ; TOTH, Thomas ; KIRDA, Engin: Service Specific Anomaly Detection for Network Intrusion Detection. In: *Proceedings of the 2002 ACM Symposium on Applied Computing*. New York, NY, USA : Association for Computing Machinery, 2002 (SAC '02), S. 201–208. – URL https://doi.org/10.1145/508791.508835. – ISBN 1581134452

[Krügel und Toth 2002] KRÜGEL, Christopher ; TOTH, Thomas: Distributed Pattern Detection for Intrusion Detection. In: *In Proceedings of the Network and Distributed System Security Symposium. Internet Society, Internet Society*, 2002

[Kumar und Vajpayee 2016] KUMAR, Shyam N. ; VAJPAYEE, Amit: A Survey on Secure Cloud: Security and Privacy in Cloud Computing. In: *American Journal of Systems and Software* 4 (2016), Nr. 1, S. 14–26. – URL http://pubs.sciepub.com/ajss/4/1/2. – ISSN 2372-7071

[Kumar u. a. 2012] KUMAR, Swarun ; SHI, Lixin ; AHMED, Nabeel ; GIL, Stephanie ; KATABI, Dina ; RUS, Daniela: CarSpeak: A Content-Centric Network for Autonomous Driving. In: *SIGCOMM Comput. Commun. Rev.* 42 (2012), August, Nr. 4, S. 259–270. – URL https://doi.org/10.1145/2377677.2377724. – ISSN 0146-4833

[Larson u. a. 2008] LARSON, U. E. ; NILSSON, D. ; JONSSON, Elias: An approach to specification-based attack detection for in-vehicle networks. In: *2008 IEEE Intelligent Vehicles Symposium* (2008), S. 220–225

[Lazarevic und Kumar 2005] LAZAREVIC, Aleksandar ; KUMAR, Vipin: Feature Bagging for Outlier Detection. In: *Proceedings of the Eleventh ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*. New York, NY, USA : Association for Computing Machinery, 2005 (KDD '05), S. 157–166. – URL https://doi.org/10.1145/1081870.1081891. – ISBN 159593135X

[Lazarevic u. a. 2005] LAZAREVIC, Aleksandar ; KUMAR, Vipin ; SRIVASTAVA, Jaideep: *Intrusion Detection: A Survey*. S. 19–78. Boston, MA : Springer US, 2005

[LeCun 1998] LECUN, Y.: The MNIST database of handwritten digits. (1998)

[Lecun u. a. 1998] LECUN, Y. ; BOTTOU, L. ; BENGIO, Y. ; HAFFNER, P.: Gradient-based learning applied to document recognition. In: *Proceedings of the IEEE* 86 (1998), Nr. 11, S. 2278–2324

[Lee u. a. 2015] LEE, H. ; CHOI, K. ; CHUNG, K. ; KIM, J. ; YIM, K.: Fuzzing CAN Packets into Automobiles. In: *2015 IEEE 29th International Conference on Advanced Information Networking and Applications*, 2015, S. 817–821

[Lee u. a. 2017] LEE, H. ; JEONG, S. H. ; KIM, H. K.: OTIDS: A Novel Intrusion Detection System for In-vehicle Network by Using Remote Frame. In: *2017 15th Annual Conference on Privacy, Security and Trust (PST)*, 2017, S. 57–5709

[Lee u. a. 2014] LEE, S. ; PARK, Y. ; LIM, H. ; SHON, T.: Study on Analysis of Security Vulnerabilities and Countermeasures in ISO/IEC 15118 Based Electric Vehicle Charging Technology. In: *2014 International Conference on IT Convergence and Security (ICITCS)*, 2014, S. 1–4

[Lee und Gerla 2010] LEE, Uichin ; GERLA, Mario: A survey of urban vehicular sensing platforms. In: *Computer Networks* 54 (2010), Nr. 4, S. 527 – 544. – URL http://www.sciencedirect.com/science/article/pii/S1389128609002382. – Advances in Wireless and Mobile Networks. – ISSN 1389-1286

[Li u. a. 2017] LI, Chunyuan ; LIU, Hao ; CHEN, Changyou ; PU, Yunchen ; CHEN, Liqun ; HENAO, Ricardo ; CARIN, Lawrence: ALICE: Towards Understanding Adversarial Learning for Joint Distribution Matching. In: *Proc. NIPS 2017*, 2017, S. 5501–5509

[Li u. a. 2019] LI, Y. ; YANG, M. ; ZHANG, Z.: A Survey of Multi-View Representation Learning. In: *IEEE Transactions on Knowledge and Data Engineering* 31 (2019), Nr. 10, S. 1863–1883

[Liang u. a. 2020] LIANG, J. ; LIN, Q. ; CHEN, J. ; ZHU, Y.: A Filter Model Based on Hidden Generalized Mixture Transition Distribution Model for Intrusion Detection System in Vehicle Ad Hoc Networks. In: *IEEE Transactions on Intelligent Transportation Systems* 21 (2020), Nr. 7, S. 2707–2722

[Liao u. a. 2013] LIAO, Hung-Jen ; RICHARD LIN, Chun-Hung ; LIN, Ying-Chih ; TUNG, Kuang-Yuan: Intrusion detection system: A comprehensive review. In: *Journal of Network and Computer Applications* 36 (2013), Nr. 1, S. 16 – 24. – URL http://www.sciencedirect.com/science/article/pii/S1084804512001944. – ISSN 1084-8045

[Linda u. a. 2011] LINDA, O. ; MANIC, M. ; VOLLMER, T. ; WRIGHT, J.: Fuzzy logic based anomaly detection for embedded network security cyber sensor. In: *2011 IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, 2011, S. 202–209

[Liu u. a. 2012] LIU, Fei T. ; TING, Kai M. ; ZHOU, Zhi-Hua: Isolation-Based Anomaly Detection. In: *ACM Trans. Knowl. Discov. Data* 6 (2012), März, Nr. 1. – URL https://doi.org/10.1145/2133360.2133363. – ISSN 1556-4681

[Liu u. a. 2017] LIU, J. ; ZHANG, S. ; SUN, W. ; SHI, Y.: In-Vehicle Network Attacks and Countermeasures: Challenges and Future Directions. In: *IEEE Network* 31 (2017), Nr. 5, S. 50–58

[Liu u. a. 2018] LIU, Q. ; LIU, T. ; LIU, Z. ; WANG, Y. ; JIN, Y. ; WEN, W.: Security analysis and enhancement of model compressed deep learning systems under adversarial attacks. In: *2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC)*, 2018, S. 721–726

[Loaiza u. a. 2019] LOAIZA, Francisco L. ; BIRDWELL, John D. ; KENNEDY, George L. ; VISSER, Dale: Utility of Artificial Intelligence and Machine Learning in Cybersecurity / Institute for Defense Analyses. URL http://www.jstor.org/stable/resrep22692, 2019. – Forschungsbericht

[Lotfi Shahreza u. a. 2011] LOTFI SHAHREZA, M. ; MOAZZAMI, D. ; MOSHIRI, B. ; DELAVAR, M.R.: Anomaly detection using a self-organizing map and particle swarm optimization. In: *Scientia Iranica* 18 (2011), Nr. 6, S. 1460–1468. – URL https://www.sciencedirect.com/science/article/pii/S1026309811001751. – ISSN 1026-3098

[Loukas und Öke 2010] LOUKAS, G. ; ÖKE, G.: Protection Against Denial of Service Attacks: A Survey. In: *The Computer Journal* 53 (2010), Nr. 7, S. 1020–1037

[Lu u. a. 2014] LU, N. ; CHENG, N. ; ZHANG, N. ; SHEN, X. ; MARK, J. W.: Connected Vehicles: Solutions and Challenges. In: *IEEE Internet of Things Journal* 1 (2014), Nr. 4, S. 289–299

[Luo und Hou 2019] LUO, Feng ; HOU, Shuo: Security Mechanisms Design of Automotive Gateway Firewall. In: *SAE Technical Paper*, SAE International, 04 2019. – URL https://doi.org/10.4271/2019-01-0481

[van der Maaten und Hinton 2008] MAATEN, Laurens van der ; HINTON, Geoffrey: Visualizing Data using t-SNE. In: *Journal of Machine Learning Research* 9 (2008), Nr. 86, S. 2579–2605. – URL http://jmlr.org/papers/v9/vandermaaten08a.html

[Mahjabin u. a. 2017] MAHJABIN, Tasnuva ; XIAO, Yang ; SUN, Guang ; JIANG, Wangdong: A survey of distributed denial-of-service attack, prevention, and mitigation techniques. In: *International Journal of Distributed Sensor Networks* 13 (2017), Nr. 12, S. 1550147717741463. – URL https://doi.org/10.1177/1550147717741463

[Majeed und Altaf 2019] MAJEED, Insha ; ALTAF, Insha: A Specification-based Intrusion Detection Model for AODV. (2019), 09

[Manchanda und Singh 2015] MANCHANDA, Kimi ; SINGH, Amarpreet: Covert Communication in VANETS using Internet Protocol Header Bit. In: *International Journal of Computer Applications* 123 (2015), S. 10–14

[Mandy und Mahgoub 2018] MANDY, C. ; MAHGOUB, I.: Implementation of the WAVE 1609.2 Security Services Standard and Encountered Issues and Challenges. In: *2018 9th IEEE Annual Ubiquitous Computing, Electronics Mobile Communication Conference (UEMCON)*, 2018

[Marchetti und Stabili 2017] MARCHETTI, M. ; STABILI, D.: Anomaly detection of CAN bus messages through analysis of ID sequences. In: *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017, S. 1577–1583

[Marcos Alvarez u. a. 2013] MARCOS ALVAREZ, Alejandro ; YAMADA, Makoto ; KIMURA, Akisato ; IWATA, Tomoharu: Clustering-Based Anomaly Detection in Multi-View Data. In: *Proceedings of the 22nd ACM International Conference on Information & Knowledge Management*. New York, NY, USA : Association for Computing Machinery, 2013 (CIKM '13), S. 1545–1548. – URL https://doi.org/10.1145/2505515.2507840. – ISBN 9781450322638

[Martinelli u. a. 2017] MARTINELLI, F. ; MERCALDO, F. ; NARDONE, V. ; SANTONE, A.: Car hacking identification through fuzzy logic algorithms. In: *2017 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2017, S. 1–7

[Matheus und Königseder 2015] MATHEUS, Kirsten ; KÖNIGSEDER, Thomas: *Automotive Ethernet*. 1st. USA : Cambridge University Press, 2015. – ISBN 1107057280

[Mattia u. a. 2019] MATTIA, Federico D. ; GALEONE, Paolo ; SIMONI, Michele D. ; GHELFI, Emanuele: A Survey on GANs for Anomaly Detection. In: *CoRR* abs/1906.11632 (2019). – URL http://arxiv.org/abs/1906.11632

[mbedlabs 2016] MBEDLABS: *Controller Area Network*. https://www.slideshare.net/mbedlabsTechnosoluti/can-bus-65612867. 2016

[McCorduck 2004] MCCORDUCK, Pamela: *Machines Who Think: A Personal Inquiry into the History and Prospects of Artificial Intelligence*. AK Peters Ltd, 2004. – ISBN 1568812051

[Mcculloch und Pitts 1943] MCCULLOCH, Warren ; PITTS, Walter: A Logical Calculus of Ideas Immanent in Nervous Activity. In: *Bulletin of Mathematical Biophysics* 5 (1943), S. 127–147

[Melnyk u. a. 2016] MELNYK, Igor ; MATTHEWS, Bryan ; VALIZADEGAN, Hamed ; BANERJEE, Arindam ; OZA, Nikunj: Vector Autoregressive Model-Based Anomaly Detection in Aviation Systems. In: *Journal of Aerospace Information Systems* 13 (2016), Nr. 4, S. 161–173. – URL https://doi.org/10.2514/1.I010394

[Miljković 2010] MILJKOVIĆ, D.: Review of novelty detection methods. In: *The 33rd International Convention MIPRO*, 2010, S. 593–598

[Miller und Valasek 2014] MILLER, Charlie ; VALASEK, Chris: *Adventures in Automotive Networks and Control Units*. 2014

[Mirkovic und Reiher 2004] MIRKOVIC, Jelena ; REIHER, Peter: A Taxonomy of DDoS Attack and DDoS Defense Mechanisms. In: *SIGCOMM Comput. Commun. Rev.* 34 (2004), April, Nr. 2, S. 39–53. – URL https://doi.org/10.1145/997150.997156. – ISSN 0146-4833

[Miyato u. a. 2018] MIYATO, Takeru ; KATAOKA, Toshiki ; KOYAMA, Masanori ; YOSHIDA, Yuichi: Spectral Normalization for Generative Adversarial Networks. In: *CoRR* abs/1802.05957 (2018)

[Moore u. a. 2017] MOORE, Michael R. ; BRIDGES, Robert A. ; COMBS, Frank L. ; STARR, Michael S. ; PROWELL, Stacy J.: Modeling Inter-Signal Arrival Times for Accurate Detection of CAN Bus Signal Injection Attacks: A Data-Driven Approach to in-Vehicle Intrusion Detection. New York, NY, USA : Association for Computing Machinery, 2017 (CISRC '17). – URL https://doi.org/10.1145/3064814.3064816. – ISBN 9781450348553

[Mukkamala u. a. 2002] MUKKAMALA, S. ; JANOSKI, G. ; SUNG, A.: Intrusion detection using neural networks and support vector machines. In: *Proceedings of the 2002 International Joint Conference on Neural Networks. IJCNN'02 (Cat. No.02CH37290)* Bd. 2, 2002, S. 1702–1707 vol.2

[Mukkamala u. a. 2005] MUKKAMALA, Srinivas ; SUNG, Andrew ; ABRAHAM, Ajith: Intrusion detection using an ensemble of intelligent paradigms. In: *Journal of Network and Computer Applications* 28 (2005), 04, S. 167–182

[Mundhenk u. a. 2015] MUNDHENK, P. ; STEINHORST, S. ; LUKASIEWYCZ, M. ; FAHMY, S. A. ; CHAKRABORTY, S.: Lightweight authentication for secure automotive networks. In: *2015 Design, Automation Test in Europe Conference Exhibition (DATE)*, 2015, S. 285–288

[Mundhenk u. a. 2017] MUNDHENK, Philipp ; PAVERD, Andrew ; MROWCA, Artur ; STEINHORST, Sebastian ; LUKASIEWYCZ, Martin ; FAHMY, Suhaib A. ; CHAKRABORTY, Samarjit: Security in Automotive Networks. In: *ACM Transactions on Design Automation of Electronic Systems* 22 (2017), Mar, Nr. 2, S. 1–27. – URL http://dx.doi.org/10.1145/2960407. – ISSN 1557-7309

[Mustafa u. a. 2013] MUSTAFA, M. A. ; ZHANG, N. ; KALOGRIDIS, G. ; FAN, Z.: Smart electric vehicle charging: Security analysis. In: *2013 IEEE PES Innovative Smart Grid Technologies Conference (ISGT)*, 2013, S. 1–6

[Münz u. a. 2007] MÜNZ, Gerhard ; LI, Sa ; CARLE, Georg: Traffic Anomaly Detection Using KMeans Clustering. In: *In GI/ITG Workshop MMBnet*, 2007

[Müter u. a. 2010] MÜTER, M. ; GROLL, A. ; FREILING, F. C.: A structured approach to anomaly detection for in-vehicle networks. In: *2010 Sixth International Conference on Information Assurance and Security*, 2010, S. 92–98

[Natale u. a. 2012] NATALE, Marco D. ; ZENG, Haibo ; GIUSTO, Paolo ; GHOSAL, Arkadeb: *Understanding and Using the Controller Area Network Communication Protocol: Theory and Practice*. Springer Publishing Company, Incorporated, 2012. – ISBN 1461403138

[Navet und Simonot-Lion 2013] NAVET, Nicolas ; SIMONOT-LION, Françoise: In-vehicle communication networks - a historical perspective and review. In: ZURAWSKI, Richard (Hrsg.): *Industrial Communication Technology Handbook, Second Edition*. CRC Press Taylor&Francis, 2013. – URL https://hal.inria.fr/hal-00876524

[Nilsson u. a. 2008] NILSSON, D. K. ; PHUNG, P. H. ; LARSON, U. E.: Vehicle ECU classification based on safety-security characteristics. In: *IET Road Transport Information and Control - RTIC 2008 and ITS United Kingdom Members' Conference*, 2008, S. 1–7

[Nilsson u. a. 2008] NILSSON, Dennis K. ; LARSON, Ulf ; PICASSO, Francesco ; JONSSON, Erland: A First Simulation of Attacks in the Automotive Network Communications Protocol FlexRay. In: *CISIS'08, Genova, Italy, October 23-24, 2008*, 2008, S. 84–91

[Northcutt 1999] NORTHCUTT, Stephen: *Network Intrusion Detection: An Analyst's Handbook*. USA : New Riders Publishing, 1999. – ISBN 0735708681

[Nowdehi u. a. 2017] NOWDEHI, N. ; LAUTENBACH, A. ; OLOVSSON, T.: In-Vehicle CAN Message Authentication: An Evaluation Based on Industrial Criteria. In: *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, 2017, S. 1–7

[Nowozin u. a. 2016] NOWOZIN, Sebastian ; CSEKE, Botond ; TOMIOKA, Ryota: F-GAN: Training Generative Neural Samplers Using Variational Divergence Minimization. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. Red Hook, NY, USA : Curran Associates Inc., 2016 (NIPS'16), S. 271–279. – ISBN 9781510838819

[nxp] NXP: *Future Vehicle Networks and ECUs*. https://www.nxp.com/docs/en/white-paper/FVNECUA4WP.pdf. – Accessed: 2020-09-30

[Olufowobi u. a. 2020] OLUFOWOBI, H. ; YOUNG, C. ; ZAMBRENO, J. ; BLOOM, G.: SAIDuCANT: Specification-Based Automotive Intrusion Detection Using Controller Area Network (CAN) Timing. In: *IEEE Transactions on Vehicular Technology* 69 (2020), Nr. 2, S. 1484–1494

[Onishi u. a. 2017] ONISHI, Hirofumi ; WU, Kelly ; YOSHIDA, Kazuo ; KATO, Takeshi: Approaches for Vehicle Cyber-Security in the US. In: *International Journal of Automotive Engineering* 8 (2017), Nr. 1, S. 1–6

[van den Oord u. a. 2016] OORD, Aäron van den ; DIELEMAN, Sander ; ZEN, Heiga ; SIMONYAN, Karen ; VINYALS, Oriol ; GRAVES, Alex ; KALCHBRENNER, Nal ; SENIOR, Andrew W. ; KAVUKCUOGLU, Koray: WaveNet: A Generative Model for Raw Audio. In: *CoRR* abs/1609.03499 (2016)

[Othmane u. a. 2015] OTHMANE, Lotfi B. ; WEFFERS, Harold ; MOHAMAD, Mohd M. ; WOLF, Marko: *A Survey of Security and Privacy in Connected Vehicles*. S. 217–247. In: BENHADDOU, Driss (Hrsg.) ; AL-FUQAHA, Ala (Hrsg.): *Wireless Sensor and Mobile Ad-Hoc Networks: Vehicular and Space Applications*. New York, NY : Springer New York, 2015. – URL https://doi.org/10.1007/978-1-4939-2468-4_10

[Ozcelik u. a. 2017] OZCELIK, M. M. ; IRMAK, E. ; OZDEMIR, S.: A hybrid trust based intrusion detection system for wireless sensor networks. In: *2017 International Symposium on Networks, Computers and Communications (ISNCC)*, 2017, S. 1–6

[Pang u. a. 2018] PANG, G. ; CAO, L. ; CHEN, L. ; LIAN, Defu ; LIU, H.: Sparse Modeling-Based Sequential Ensemble Learning for Effective Outlier Detection in High-Dimensional Numeric Data. In: *AAAI*, 2018

[Pang u. a. 2017] PANG, Guansong ; CAO, Longbing ; CHEN, Ling ; LIU, Huan: Learning Homophily Couplings from Non-IID Data for Joint Feature Selection and Noise-Resilient Outlier Detection. In: *Proceedings of the*

Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17, URL https://doi.org/10.24963/ijcai.2017/360, 2017, S. 2585–2591

[Pang u. a. 2021] PANG, Guansong ; SHEN, Chunhua ; CAO, Longbing ; HENGEL, Anton Van D.: Deep Learning for Anomaly Detection: A Review. In: *ACM Comput. Surv.* 54 (2021), März, Nr. 2. – URL https://doi.org/10.1145/3439950. – ISSN 0360-0300

[Papalexakis u. a. 2012] PAPALEXAKIS, E. E. ; BEUTEL, A. ; STEENKISTE, P.: Network Anomaly Detection Using Co-clustering. In: *2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, 2012, S. 403–410

[Patil und Kamble 2018] PATIL, H. A. ; KAMBLE, M. R.: A Survey on Replay Attack Detection for Automatic Speaker Verification (ASV) System. In: *2018 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2018, S. 1047–1053

[Pawelec u. a. 2019] PAWELEC, Krzysztof ; BRIDGES, Robert A. ; COMBS, Frank L.: Towards a CAN IDS Based on a Neural Network Data Field Predictor. In: *Proceedings of the ACM Workshop on Automotive Cybersecurity*. New York, NY, USA : Association for Computing Machinery, 2019 (AutoSec '19), S. 31–34. – URL https://doi.org/10.1145/3309171.3309180. – ISBN 9781450361804

[Peng u. a. 2007] PENG, Tao ; LECKIE, Christopher ; RAMAMOHANARAO, Kotagiri: Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems. In: *ACM Comput. Surv.* 39 (2007), April, Nr. 1, S. 3–es. – URL https://doi.org/10.1145/1216370.1216373. – ISSN 0360-0300

[Peng u. a. 2018] PENG, Zhen ; LUO, Minnan ; LI, Jundong ; LIU, Huan ; ZHENG, Qinghua: ANOMALOUS: A Joint Modeling Approach for Anomaly Detection on Attributed Networks. In: *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, International Joint Conferences on Artificial Intelligence Organization, 7 2018, S. 3513–3519. – URL https://doi.org/10.24963/ijcai.2018/488

[Petit u. a. 2015] PETIT, J. ; STOTTELAAR, Bas ; FEIRI, M.: Remote Attacks on Automated Vehicles Sensors : Experiments on Camera and LiDAR, 2015

[Petit und Shladover 2014] PETIT, Jonathan ; SHLADOVER, Steven: Potential Cyberattacks on Automated Vehicles. In: *Intelligent Transportation Systems, IEEE Transactions on* PP (2014), 09, S. 1–11

[Philip Baczwewski 2000] PHILIP BACZWEWSKI, Michael M.: *Web-based Mail Issues*. S. 119 – 145. In: SYNGRESS (Hrsg.): *E-Mail Virus Protection Handbook*. Burlington : Syngress, 2000. – URL http://www.sciencedirect.com/science/article/pii/B9781928994237500082. – ISBN 978-1-928994-23-7

[Pimentel u. a. 2014] PIMENTEL, Marco A. ; CLIFTON, David A. ; CLIFTON, Lei ; TARASSENKO, Lionel: A review of novelty detection. In: *Signal Processing* 99 (2014), S. 215–249. – URL https://www.sciencedirect.com/science/article/pii/S016516841300515X. – ISSN 0165-1684

[Poudel und Munir 2021] POUDEL, B. ; MUNIR, A.: Design and Evaluation of a Reconfigurable ECU Architecture for Secure and Dependable Automotive CPS. In: *IEEE Transactions on Dependable and Secure Computing* 18 (2021), S. 235–252

[Prowell u. a. 2010] PROWELL, Stacy ; KRAUS, Rob ; BORKIN, Mike: CHAPTER 6 - Man-in-the-Middle. In: PROWELL, Stacy (Hrsg.) ; KRAUS, Rob (Hrsg.) ; BORKIN, Mike (Hrsg.): *Seven Deadliest Network Attacks*. Boston : Syngress, 2010, S. 101 – 120. – URL http://www.sciencedirect.com/science/article/pii/B9781597495493000067. – ISBN 978-1-59749-549-3

[Qiang Hu 2018] QIANG HU, Qiang H.: Review of Secure Communication Approaches for In-Vehicle Network. In: *International Journal of Automotive Technology* (2018)

[Radford u. a. 2016] RADFORD, Alec ; METZ, Luke ; CHINTALA, Soumith: Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks. In: *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, URL http://arxiv.org/abs/1511.06434, 2016

[Radu und Garcia 2016] RADU, Andreea-Ina ; GARCIA, Flavio D.: LeiA: A Lightweight Authentication Protocol for CAN. In: ASKOXYLAKIS, Ioannis (Hrsg.) ; IOANNIDIS, Sotiris (Hrsg.) ; KATSIKAS, Sokratis (Hrsg.) ; MEADOWS, Catherine (Hrsg.): *Computer Security – ESORICS 2016*. Cham : Springer International Publishing, 2016, S. 283–300

[Robinson-Mallett und Hansack 2015] ROBINSON-MALLETT, Christopher ; HANSACK, Sebastian: A Model of an Automotive Security Concept Phase. In: *Proceedings of the 10th Annual Cyber and Information Security Research Conference*. New York, NY, USA : Association for Computing Machinery, 2015 (CISR '15). – URL https://doi.org/10.1145/2746266.2746282. – ISBN 9781450333450

[Rong u. a. 2013] RONG, Chunming ; ZHAO, Gansen ; YAN, Liang ; CAYIRCI, Erdal ; CHENG, Hongbing: Chapter 21 - Radio Frequency Identification Security. In: VACCA, John R. (Hrsg.): *Computer and Information Security Handbook (Third Edition)*. Third Edition. Boston : Morgan Kaufmann, 2013, S. 369 – 385. – URL http://www.sciencedirect.com/science/article/pii/B9780128038437000211. – ISBN 978-0-12-803843-7

[Rouf u. a. 2010] ROUF, Ishtiaq ; MILLER, Robert ; MUSTAFA, Hossen ; TAYLOR, Travis ; OH, Sangho ; XU, Wenyuan ; GRUTESER, Marco ; TRAPPE, W. ; SESKAR, Ivan: Security and Privacy Vulnerabilities of In-Car Wireless Networks: A Tire Pressure Monitoring System Case Study, 08 2010, S. 323–338

[Rumelhart u. a. 1986] RUMELHART, D. E. ; HINTON, G. E. ; WILLIAMS, R. J.: *Learning Internal Representations by Error Propagation*. S. 318–362. In: *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1: Foundations*. Cambridge, MA, USA : MIT Press, 1986. – ISBN 026268053X

[Sabokrou u. a. 2018] SABOKROU, Mohammad ; KHALOOEI, Mohammad ; FATHY, Mahmood ; ADELI, Ehsan: Adversarially Learned One-Class Classifier for Novelty Detection. In: *Proc. CVPR 2018*, 2018, S. 3379–3388

[Safavian und Landgrebe 1991] SAFAVIAN, S. R. ; LANDGREBE, D.: A survey of decision tree classifier methodology. In: *IEEE Transactions on Systems, Man, and Cybernetics* 21 (1991), Nr. 3, S. 660–674

[Salimans u. a. 2016] SALIMANS, Tim ; GOODFELLOW, Ian ; ZAREMBA, Wojciech ; CHEUNG, Vicki ; RADFORD, Alec ; CHEN, Xi: Improved Techniques for Training GANs. In: *Proceedings of the 30th International Conference on Neural Information Processing Systems*. Red Hook, NY, USA : Curran Associates Inc., 2016 (NIPS'16), S. 2234–2242. – ISBN 9781510838819

[Saxe u. a. 2019] SAXE, Andrew ; BANSAL, Yamini ; DAPELLO, Joel ; ADVANI, Madhu ; KOLCHINSKY, Artemy ; TRACEY, Brendan ; COX, David: On the information bottleneck theory of deep learning. In: *Journal of Statistical Mechanics: Theory and Experiment* 2019 (2019), 12, S. 124020

[Schlegl u. a. 2017] SCHLEGL, Thomas ; SEEBÖCK, Philipp ; WALDSTEIN, Sebastian M. ; SCHMIDT-ERFURTH, Ursula ; LANGS, Georg: Unsupervised Anomaly Detection with Generative Adversarial Networks to Guide Marker Discovery. In: *Information Processing in Medical Imaging*. Cham : Springer International Publishing, 2017, S. 146–157. – ISBN 978-3-319-59050-9

[Schlegl u. a. 2019] SCHLEGL, Thomas ; SEEBÖCK, Philipp ; WALDSTEIN, Sebastian M. ; LANGS, Georg ; SCHMIDT-ERFURTH, Ursula: f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks. In: *Medical Image Analysis* 54 (2019), S. 30–44. – URL https://www.sciencedirect.com/science/article/pii/S1361841518302640. – ISSN 1361-8415

[Schneier 1995] SCHNEIER, Bruce: *Applied Cryptography (2nd Ed.): Protocols, Algorithms, and Source Code in C*. USA : John Wiley & Sons, Inc., 1995. – ISBN 0471117099

[Schölkopf u. a. 1997] SCHÖLKOPF, Bernhard ; SMOLA, Alexander ; MÜLLER, Klaus-Robert: Kernel principal component analysis. In: GERSTNER, Wulfram (Hrsg.) ; GERMOND, Alain (Hrsg.) ; HASLER, Martin (Hrsg.) ; NICOUD, Jean-Daniel (Hrsg.): *Artificial Neural Networks — ICANN'97*. Berlin, Heidelberg : Springer Berlin Heidelberg, 1997, S. 583–588. – ISBN 978-3-540-69620-9

[Schölkopf u. a. 1999] SCHÖLKOPF, Bernhard ; WILLIAMSON, Robert ; SMOLA, Alex ; SHAWE-TAYLOR, John ; PLATT, John: Support Vector Method for Novelty Detection. In: *Proceedings of the 12th International Conference on Neural Information Processing Systems*. Cambridge, MA, USA : MIT Press, 1999 (NIPS'99), S. 582–588

[Schölkopf u. a. 2002] SCHÖLKOPF, Bernhard ; SMOLA, Alex ; SMOLA, Alexander ; SMOLA, A: Support Vector Machines and Kernel Algorithms. In: *Encyclopedia of Biostatistics, 5328-5335 (2005)* (2002), 04

[Sedjelmaci u. a. 2019] SEDJELMACI, H. ; HADJI, M. ; ANSARI, N.: Cyber Security Game for Intelligent Transportation Systems. In: *IEEE Network* 33 (2019), Nr. 4, S. 216–222

[Seifert und Obermaisser 2014] SEIFERT, S. ; OBERMAISSER, R.: Secure automotive gateway — Secure communication for future cars. In: *2014 12th IEEE International Conference on Industrial Informatics (INDIN)*, 2014, S. 213–220

[Sen Nie 2017] SEN NIE, Yuefeng DuLand ing L.: *Free-Fall: Hacking Tesla from Wireless to CAN Bus*. 2017

[Seo u. a. 2018] SEO, E. ; SONG, H. M. ; KIM, H. K.: GIDS: GAN based Intrusion Detection System for In-Vehicle Network. In: *2018 16th Annual Conference on Privacy, Security and Trust (PST)*, 2018, S. 1–6

[Seri und Vishnepolsky] SERI ; VISHNEPOLSKY, Gregory: *BLUEBORNE ON ANDROID Exploiting an RCE Over the AirBen*

[Shavit u. a. 2007] SHAVIT, Moshe ; GRYC, Andy ; MIUCIC, Radovan: Firmware Update Over The Air (FOTA) for Automotive Industry. In: *SAE Technical Paper*, SAE International, 08 2007. – URL https://doi.org/10.4271/2007-01-3523

[Sheehan u. a. 2019] SHEEHAN, Barry ; MURPHY, Finbarr ; MULLINS, Martin ; RYAN, Cian: Connected and autonomous vehicles: A cyber-risk classification framework. In: *Transportation Research Part A: Policy and Practice* 124 (2019), S. 523–536. – URL https://www.sciencedirect.com/science/article/pii/S096585641830555X. – ISSN 0965-8564

[Smith 2015] SMITH: Car Hacker's Handbook. (2015)

[Song u. a. 2020] SONG, Hyun M. ; WOO, Jiyoung ; KIM, Huy K.: In-vehicle network intrusion detection using deep convolutional neural network. In: *Vehicular Communications* (2020)

[Song u. a. 2007] SONG, Xiuyao ; WU, Mingxi ; JERMAINE, Christopher ; RANKA, Sanjay: Conditional Anomaly Detection. In: *Knowledge and Data Engineering, IEEE Transactions on* 19 (2007), 06, S. 631–645

[Srivastava und Salakhutdinov 2014] SRIVASTAVA, Nitish ; SALAKHUTDINOV, Ruslan: Multimodal Learning with Deep Boltzmann Machines. In: *J. Mach. Learn. Res.* 15 (2014), Januar, Nr. 1, S. 2949–2980. – ISSN 1532-4435

[Sun u. a. 2016] SUN, S. ; HU, J. ; PENG, Y. ; PAN, X. ; ZHAO, L. ; FANG, J.: Support for vehicle-to-everything services based on LTE. In: *IEEE Wireless Communications* 23 (2016), Nr. 3, S. 4–8

[Sun u. a. 2015] SUN, X. ; XIA, L. ; JIA, S.: Enhancing location privacy for electric vehicles by obfuscating the linkages of charging events. In: *2015 IEEE 5th International Conference on Electronics Information and Emergency Communication*, 2015, S. 155–158

[Syarif u. a. 2012] SYARIF, Iwan ; PRUGEL-BENNETT, Adam ; WILLS, Gary: Unsupervised Clustering Approach for Network Anomaly Detection. In: *Networked Digital Technologies*. Berlin, Heidelberg : Springer Berlin Heidelberg, 2012, S. 135–145

[Szydlowski 1992] SZYDLOWSKI, Craig P.: CAN Specification 2.0: Protocol and Implementations. In: *SAE Technical Paper*, SAE International, 08 1992. – URL https://doi.org/10.4271/921603

[Tanenbaum und van Steen 2002] TANENBAUM, Andrew S. ; STEEN, Maarten van: *Distributed Systems: Principles and Paradigms*. Pearson Prentice Hall, 2002

[Tanksale 2019] TANKSALE, V.: Intrusion Detection For Controller Area Network Using Support Vector Machines. In: *2019 IEEE 16th International Conference on Mobile Ad Hoc and Sensor Systems Workshops (MASSW)*, 2019, S. 121–126

[Tashiro u. a. 2017] TASHIRO, A. ; MURAOKA, H. ; ARAKI, S. ; KAKIZAKI, K. ; UEHARA, S.: A secure protocol consisting of two different security-level message authentications over CAN. In: *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*, 2017, S. 1520–1524

[Taylor u. a. 2015] TAYLOR, A. ; JAPKOWICZ, N. ; LEBLANC, S.: Frequency-based anomaly detection for the automotive CAN bus. In: *2015 World Congress on Industrial Control Systems Security (WCICSS)*, 2015, S. 45–49

[pico technology] TECHNOLOGY pico: *CAN and CAN FD bus decoding*. https://www.picoauto.com/library/picoscope/can-bus-serial-protocol-decoding. – Accessed: 2020-09-30

[Thing und Wu 2016] THING, V. L. L. ; WU, J.: Autonomous Vehicle Security: A Taxonomy of Attacks and Defences. In: *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2016, S. 164–170

[Tomlinson u. a. 2018a] TOMLINSON, A. ; BRYANS, J. ; SHAIKH, S. A. ; KALUTARAGE, H. K.: Detection of Automotive CAN Cyber-Attacks by Identifying Packet Timing Anomalies in Time Windows. In: *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, 2018, S. 231–238

[Tomlinson u. a. 2018b] TOMLINSON, A. ; BRYANS, J. ; SHAIKH, S. A. ; KALUTARAGE, H. K.: Detection of Automotive CAN Cyber-Attacks by Identifying Packet Timing Anomalies in Time Windows. In: *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, 2018, S. 231–238

[Tomlinson u. a. 2018c] TOMLINSON, A. ; BRYANS, J. ; SHAIKH, S. A. ; KALUTARAGE, H. K.: Detection of Automotive CAN Cyber-Attacks by Identifying Packet Timing Anomalies in Time Windows. In: *2018 48th Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, 2018, S. 231–238

[Toyama u. a. 2018] TOYAMA, Tsuyoshi ; YOSHIDA, Takuya ; OGUMA, Hisashi ; MATSUMOTO, Tsutomu: PASTA : Portable Automotive Security Testbed with Adaptability, 2018

[Tsai u. a. 2009] TSAI, Chih-Fong ; HSU, Yu-Feng ; LIN, Chia-Ying ; LIN, Wei-Yang: Intrusion detection by machine learning: A review. In: *Expert Systems with Applications* 36 (2009), Nr. 10, S. 11994 – 12000. – URL http://www.sciencedirect.com/science/article/pii/S0957417409004801. – ISSN 0957-4174

[Vaidya und Mouftah 2018] VAIDYA, B. ; MOUFTAH, H. T.: Deployment of Secure EV Charging System Using Open Charge Point Protocol. In: *2018 14th International Wireless Communications Mobile Computing Conference (IWCMC)*, 2018, S. 922–927

[Veeramrddy u. a. 2011] VEERAMRDDY, Jyothsna ; PRASAD, V. ; PRASAD, Koneti: A Review of Anomaly based Intrusion Detection Systems. In: *International Journal of Computer Applications* 28 (2011), 08, S. 26–35

[Venkataramanan u. a. 2019] VENKATARAMANAN, Shashanka ; PENG, Kuan-Chuan ; SINGH, Rajat V. ; MAHALANOBIS, Abhijit: Attention Guided Anomaly Detection and Localization in Images. In: *CoRR* abs/1911.08616 (2019). – URL http://arxiv.org/abs/1911.08616

[Veropoulos u. a. 1999] VEROPOULOS, K. ; CAMPBELL, C. ; CRISTIANINI, N.: Controlling the Sensitivity of Support Vector Machines. In: *Proc. IJCAI 1999* (1999)

[Vondrick u. a. 2016] VONDRICK, Carl ; PIRSIAVASH, Hamed ; TORRALBA, Antonio: Generating Videos with Scene Dynamics. In: *CoRR* abs/1609.02612 (2016). – URL http://arxiv.org/abs/1609.02612

[Vuong u. a. 2015] VUONG, T. P. ; LOUKAS, G. ; GAN, D.: Performance Evaluation of Cyber-Physical Intrusion Detection on a Robotic Vehicle. In: *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing*, 2015, S. 2106–2113

[Wang u. a. 2018] WANG, Chundong ; ZHAO, Zhentang ; GONG, Liangyi ; ZHU, Likun ; LIU, Zheli ; CHENG, Xiaochun: A Distributed Anomaly Detection System for In-Vehicle Network using HTM. In: *IEEE Access* PP (2018), 01, S. 1–1

[Wang u. a. 2015] WANG, Weiran ; ARORA, Raman ; LIVESCU, Karen ; BILMES, Jeff: On Deep Multi-View Representation Learning. In: *Proceedings of the 32nd International Conference on International Conference on Machine Learning - Volume 37*, JMLR.org, 2015 (ICML'15), S. 1083–1092

[Weber u. a. 2018] WEBER, Marc ; KLUG, Simon ; SAX, Eric ; ZIMMER, Bastian: Embedded Hybrid Anomaly Detection for Automotive CAN Communication. In: *9th European Congress on Embedded Real Time Software and Systems (ERTS 2018)*. Toulouse, France, Januar 2018. – URL https://hal.archives-ouvertes.fr/hal-01716805

[Williams u. a. 2002] WILLIAMS, Graham ; BAXTER, Rohan ; HE, Hongxing ; HAWKINS, Simon ; GU, Lifang: A Comparative Study of RNN for Outlier Detection in Data Mining. In: *Proceedings of the 2002 IEEE International Conference on Data Mining*. USA : IEEE Computer Society, 2002 (ICDM '02), S. 709. – ISBN 0769517544

[Wolf 2009] WOLF, Marko: *Security Engineering for Vehicular IT Systems*. 01 2009. – ISBN 978-3-8348-0795-3

[Wu und Chang 2003] WU, Gang ; CHANG, Edward Y.: Adaptive Feature-Space Conformal Transformation for Imbalanced-Data Learning. In: *Proc. ICML 2003*, 2003, S. 816–823

[Wu und Banzhaf 2010] WU, Shelly X. ; BANZHAF, Wolfgang: Review: The Use of Computational Intelligence in Intrusion Detection Systems: A Review. In: *Appl. Soft Comput.* 10 (2010), Januar, Nr. 1, S. 1–35. – URL https://doi.org/10.1016/j.asoc.2009.06.019. – ISSN 1568-4946

[Wu und Yen 2009] WU, Su-Yun ; YEN, Ester: Data mining-based intrusion detectors. In: *Expert Systems with Applications* 36 (2009), Nr. 3, Part 1, S. 5605–5612. – URL https://www.sciencedirect.com/science/article/pii/S0957417408004089. – ISSN 0957-4174

[Xu u. a. 2013] XU, Chang ; TAO, D. ; XU, Chao: A Survey on Multi-view Learning. In: *ArXiv* abs/1304.5634 (2013)

[Yang u. a. 2019] YANG, L. ; MOUBAYED, A. ; HAMIEH, I. ; SHAMI, A.: Tree-Based Intelligent Intrusion Detection System in Internet of Vehicles. In: *2019 IEEE Global Communications Conference (GLOBECOM)*, 2019, S. 1–6

[Ye und Chen 2001] YE, Nong ; CHEN, Qiang: An anomaly detection technique based on a chi-square statistic for detecting intrusions into information systems. In: *Quality and Reliability Engineering International* 17 (2001), Nr. 2, S. 105–112. – URL https://onlinelibrary.wiley.com/doi/abs/10.1002/qre.392

[Young u. a. 2019] YOUNG, Clinton ; OLUFOWOBI, Habeeb ; BLOOM, Gedare ; ZAMBRENO, Joseph: Automotive Intrusion Detection Based on Constant CAN Message Frequencies Across Vehicle Driving Modes, 2019 (AutoSec)

[Zenati u. a. 2018a] ZENATI, Houssam ; FOO, Chuan S. ; LECOUAT, Bruno ; MANEK, Gaurav ; CHANDRASEKHAR, Vijay R.: Efficient GAN-Based Anomaly Detection. In: *CoRR* abs/1802.06222 (2018)

[Zenati u. a. 2018b] ZENATI, Houssam ; ROMAIN, Manon ; FOO, Chuan-Sheng ; LECOUAT, Bruno ; CHANDRASEKHAR, Vijay: Adversarially Learned Anomaly Detection. In: *Proc. ICDM 2018*, 2018, S. 727–736

[Zeng u. a. 2016] ZENG, W. ; KHALID, M. A. S. ; CHOWDHURY, S.: In-Vehicle Networks Outlook: Achievements and Challenges. In: *IEEE Communications Surveys Tutorials* 18 (2016), Nr. 3, S. 1552–1571

[Zhang u. a. 2014] ZHANG, T. ; ANTUNES, H. ; AGGARWAL, S.: Defending Connected Vehicles Against Malware: Challenges and a Solution Framework. In: *IEEE Internet of Things Journal* 1 (2014), Nr. 1, S. 10–21

[Zhang u. a. 2020] ZHANG, Xinyang ; WANG, Ningfei ; SHEN, Hua ; JI, Shouling ; LUO, Xiapu ; WANG, Ting: Interpretable Deep Learning under Fire. In: *29th USENIX Security Symposium (USENIX Security 20)*, USENIX Association, August 2020, S. 1659–1676. – URL https://www.usenix.org/conference/usenixsecurity20/presentation/zhang-xinyang. – ISBN 978-1-939133-17-5

[Zhang u. a. 2021] ZHANG, Yu ; TIŇO, Peter ; LEONARDIS, Aleš ; TANG, Ke: A Survey on Neural Network Interpretability. (2021)

[Zhou u. a. 2015] ZHOU, C. ; HUANG, S. ; XIONG, N. ; YANG, S. ; LI, H. ; QIN, Y. ; LI, X.: Design and Analysis of Multimodel-Based Anomaly Intrusion Detection Systems in Industrial Process Automation. In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 45 (2015), Nr. 10, S. 1345–1360

[Zhu und Sastry 2011] ZHU, B. ; SASTRY, S.: Revisit Dynamic ARIMA Based Anomaly Detection. In: *2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing*, 2011, S. 1263–1268

[Zhu u. a. 2017] ZHU, J. ; PARK, T. ; ISOLA, P. ; EFROS, A. A.: Unpaired Image-to-Image Translation Using Cycle-Consistent Adversarial Networks. In: *2017 IEEE International Conference on Computer Vision (ICCV)*, 2017, S. 2242–2251

ÉCOLE DOCTORALE
Sciences et technologies de l'information et de la communication (STIC)

Titre : Apprentissage automatique pour les systèmes de détection d'intrusion dans les transports autonomes

Mots clés : Apprentissage profond, Réseaux antagonistes, Détection d'anomalies, Séries temporelle, systèmes de détection d'intrusion, réseaux véhiculaires, modèle distribué.

Résumé : De nombreuse avancée et innovations technologiques sont introduites dans le monde de l'automobile. Plusieurs domaines scientifiques et applicatifs contribuent à l'amélioration de ces avancées. L'une des dimensions importantes est la cybersécurité. Effectivement, les véhicules autonomes seront sujets aux cyberattaques et les cybers criminels pourraient pirater les systèmes d'exploitation des véhicules et perturber leur fonctionnement et mettre en danger la sûreté des passagers. Ainsi, la cybersécurité reste un obstacle à surmonter pour sécuriser les véhicules et permettre aux innovations technologiques dans le domaine des transports d'apporter des solutions aux problèmes de la société et éviter leur détournement à des fins malicieuses. En effet, la conception actuelle et future des véhicules autonomes implique de nombreuses interfaces de communication, la communication dans le véhicule entre les différents systèmes embarqués, les communications Vehicle-to-X (V2X) entre le véhicule et d'autres véhicules et structures connectés sur les routes.

Plusieurs mécanismes de défense sont implémentés pour répondre aux normes de sécurité (antivirus, pare-feu, etc.), mais nous ne pouvons pas être sûrs que toutes les failles possibles sont couvertes, spécialement dans des systèmes complexes comme les voitures autonomes. Le système de détection d'intrusion a été introduit dans le monde IT pour évaluer l'état du réseau et détecter tous les comportements malveillants. Le monde l'IT a connu beaucoup plus d'expérience en termes de mécanisme de défense qui peut inspirer la cybersécurité des transports intelligent (voiture autonome), néanmoins, ces dernières requirent leurs propres besoins et contraintes liées à la sûreté et aussi à leur architecture system. L'état actuel de l'évolution des véhicules a été rendu possible grâce à des innovations successives dans de nombreux domaines industriels et de recherche.

L'intelligence artificielle en fait partie, ses différentes techniques permettent d'apprendre et de mettre en œuvre des tâches complexe tel que la conduite autonome. Cette thèse vise à développer un system intelligent de détection d'intrusion en utilisant l'apprentissage automatique dans un contexte automobile. L'objectif est d'évaluer l'impact de l'apprentissage automatique sur l'amélioration de la sécurité des véhicules futurs (autonomes). Notre objectif principal est la sécurité des communications entres les différents systèmes dans la voiture. Dans ce but, nous menons une enquête empirique pour déterminer les besoins sous-jacents et les contraintes qu'exigent les systèmes embarqués. Nous passons en revue la littérature d'apprentissage profond pour la détection d'anomalie, on note qu'il y a un manque d'étude personnalisée sur le système de détection d'intrusion de véhicule autonome utilisant l'apprentissage profond. Dans de telles applications, les données sont déséquilibrées : le taux d'exemples normal est beaucoup plus élevé que les exemples anormaux. L'émergence du réseau antagoniste (GAN) a récemment apporté de nouveaux algorithmes pour la détection des anomalies. Nous développons une approche antagoniste (adversarial) pour la détection des anomalies, basée sur un Encoding Adversarial Network (EAN).

L'architecture future des réseaux embarqués dans les véhicules est composée de différents sous-systèmes. Chaque sous-système est responsable de services spécifiques qui assurent le fonctionnement autonome du véhicule. Pour des raisons fonctionnelles et de sécurité, les sous-systèmes sont isolés, formant une architecture de communication hiérarchiques de l'ensemble du système. Dans cette thèse, nous concevons un IDS distribué qui s'adapte à l'architecture embarquée et à ses contraintes et réduit le taux de surcharge de communication induit par le traitement de l'IDS.

Maison du doctorat de l'Université Paris-Saclay
2ème étage aile ouest, Ecole normale supérieure Paris-Saclay
4 avenue des Sciences,
91190 Gif sur Yvette, France

Title : Machine Learning for intrusion detection systems in autonomous transportation

Keywords : Deep Learning, Adversarial learning, Anomaly detection, Time series, Intrusion detection systems, In-vehicle network, Distributed models.

Abstract : Despite all the different technological innovations and advances in the automotive field, autonomous vehicles are still in the testing phase. Many actors are working on several improvements in many domains to make autonomous cars the safest option. One of the important dimensions is cybersecurity. Autonomous vehicles will be prone to cyberattacks, and criminals might be motivated to hack into the vehicles' operating systems, steal essential passenger data, or disrupt its operation and jeopardize the passenger's safety. Thus, cybersecurity remains one of the biggest obstacles to overcome to ensure vehicles safety and the contribution that this technology can bring to society.

Indeed, the actual and future design and implementation of Autonomous Vehicles imply many communication interfaces, In-vehicle communication of the embedded system, Vehicle-to-X (V2X) communications between the vehicle and other connected vehicles and structures on the roads. Even though the cybersecurity aspect is incorporated by design, meaning that the system needs to satisfy security standards (anti-virus, firewall, etc.), we cannot ensure that all possible breaches are covered. The Intrusion Detection System (IDS) has been introduced in the IT world to assess the state of the network and detect if a violation occurs. Many experiences and the history of IT have inspired the cybersecurity for autonomous vehicles. Nevertheless, autonomous vehicles exhibit their own needs and constraints.

The current state of vehicles evolution has been made possible through successive innovations in many industrial and research fields. Artificial Intelligence (AI) is one of them. It enables learning and implementing the most fundamental self-driving tasks. This thesis aims to develop an intelligent in-vehicle Intrusion detection system (IDS) using machine learning (ml) from an automotive perspective, to assess and evaluate the impact of machine learning on enhancing the security of future vehicle intrusion detection system that fits in-vehicle computational constraints.

Future In-vehicle network architecture is composed of different subsystems formed of other ECUs (Electronic Controller Units). Each subsystem is vehicles. Our primary focus is on In-vehicle communication security. We conduct an empirical investigation to determine the underlying needs and constraints that in-vehicle systems require. First, we review the deep learning literature for anomaly detection and studies on autonomous vehicle intrusion detection systems using deep learning. We notice many works on in-vehicle intrusion detection systems, but not all of them consider the constraints of autonomous vehicle systems.

We conduct an empirical investigation to determine the underlying needs and constraints that in-vehicle systems require. We review the deep learning literature for anomaly detection, and there is a lack of tailored study on autonomous vehicle intrusion detection systems using Deep Learning (DL).

In such applications, the data is unbalanced: the rate of normal examples is much higher than the anomalous examples. The emergence of generative adversarial networks (GANs) has recently brought new algorithms for anomaly detection. We develop an adversarial approach for anomaly detection based on an Encoding adversarial network (EAN). Considering the behaviour and the lightweight nature of in-vehicle networks, we show that EAN remains robust to the increase of normal examples modalities, and only a sub-part of the neural network is used for the detection phase.

Controller Area Network (CAN) is one of the most used vehicle bus standards designed to allow microcontrollers and devices to communicate. We propose a Deep CAN intrusion detection system framework. We introduce a Multi-Variate Time Series representation for asynchronous CAN data. We show that this representation enhances the temporal modelling of deep learning architectures for anomaly detection. We study different deep learning tasks (supervised/unsupervised) and compare several architectures to design an in-

Maison du doctorat de l'Université Paris-Saclay
2ème étage aile ouest, Ecole normale supérieure Paris-Saclay
4 avenue des Sciences,
91190 Gif sur Yvette, France

responsible for specific services that ensure the autonomous functioning of the vehicle. For functional and security reasons, separate subsystems are isolated, forming a hierarchical architecture of the system. In this thesis, we design a Distributed IDS that fit this in-vehicle architecture system and its constraints and reduces the communication overhead rate induced by the IDS processing.

Maison du doctorat de l'Université Paris-Saclay
2ème étage aile ouest, Ecole normale supérieure Paris-Saclay
4 avenue des Sciences,
91190 Gif sur Yvette, France