University of Nottingham
UK | CHINA | MALAYSIA

A Machine Learning-based Anomaly Detection Framework for Connected and Autonomous Vehicles Cyber Security

Qiyi He

Thesis submitted to the University of Nottingham
for the degree of Doctor of Philosophy

September 2021

I dedicated this thesis to all the members of my family: My husband, Dr Lu Dong; My father, Baoguo He; My mother, Lihua He.

Abstract

Connected and Autonomous Vehicles (CAVs) have expanded fast in recent years and have started to affect people's daily lives. It is believed that CAVs could bring benefits, including improving traffic efficiency, reducing accidents and emissions. However, the development of CAVs still faces several technical and social challenges. Issues such as cyber security have become predominant, forming an essential part of the complications of CAV deployment. The increasing number of autonomous and connected functions, however, means that CAVs are exposed to more cyber security vulnerabilities. Unlike computer cyber security attacks, cyber attacks on CAVs could lead to not only information leakage but also physical damage. According to the UK CAV Cyber Security Principles released in 2017, preventing CAVs from cyber security attacks needs to be considered at the beginning of CAV development. There is, however, no universally agreed upon or recognized framework for Connected and Autonomous Vehicles Cyber Security (CAVCS). The main aim of this thesis is to develop a machine learning-based anomaly detection CAVCS framework to detect potential attacks on CAVs.

In the thesis, new CAVCS terminology is defined to establish the theoretical foundation of the framework. A large set of potential attacks within the framework, are then investigated and evaluated based on a newly proposed severity assessment method from the aspects of target assets, risks and consequences. The severity assessment results show that the DoS attack and Fuzzy attack are the most severe cyber attacks among all the defined attacks. Besides, this specific CAVCS assessment method could be extendable for evolving technologies applied to CAVs in the future.

Based on the assessed potential attacks, four new CAVCS data sets are simulated and collected in the thesis. CAV-KDD data set generated from cyber security benchmark KDD99 covers potential attacks to inter-vehicle communications; Simulated Simu-CAN data set and real world KCAN-CAV data set cover DoS attacks and Fuzzy attacks on in-vehicle communication; Self-collected CAV-RW data set addresses the limitations of these three data sets, restoring the DoS and Fuzzy attacks scenarios to the greatest extent in the real world. In addition to filling the research gap of lacking CAVCS data sets, the four new data sets could help build machine learning models and thus provide secured CAVs in simulated environments and real world usage.

To build and assess the performance of machine learning-based anomaly detection on CAVCS, classifiers Decision Tree and Naive Bayes are introduced to each of the four new data sets. The comparison is made based on specific metrics, including accuracy, false positive rate and runtime. The results indicate that machine learning models could help to detect attacks on CAVs, among which the Decision Tree model is superior to the Naive Bayes model. However, the runtime is not sufficient for dynamic driving environments, further

enhanced performance improvements are needed on machine learning models.

To shorten the runtime without negative impacts on accuracy, feature selection methods, including Info Gain, Gain Ratio, CFS (Correlation-based Feature Selection), and Pearson method, are then adapted to machine learning models. Experimental results show that feature selection methods improve the performance of models on inter-vehicle communication data set CAV-KDD significantly due to a large number of features. While in other data sets, the impact of feature selection is not evident due to the limited number of features. Among all these, the Decision Tree remains to be the best-performance detection model. The most important attributes towards different attacks in different situations are also suggested to provide guidance for further research.

Thus, the thesis builds a new machine learning-based anomaly detection framework for CAVCS, which incorporates severity assessment, data collection, attacks detection and performance improvement. The framework has been adapted and validated in the simulated and real world environments, showing it to be effective for assessing and detecting CAV cyber attacks. The framework could provide guidance and a baseline for further CAVCS researches.

Acknowledgements

During my PhD study, I met lots of people to help and support me. I would like to express my sincere appreciation to all of you.

Firstly, I would like to thank my esteemed supervisors – Prof Xiaolin Meng and Dr Rong Qu, for their continuous support and supervision. Their immense knowledge and experience have helped me in both academic research and daily life. Prof Meng was always enthusiastic to help and provided insightful discussions about my research. He also offered me invaluable opportunities to exchange my PhD project ideas with colleagues from different organizations through seminars and international conferences. Without his help, the research could not have gone on smoothly. Dr Rong Qu patiently provided guidance and was always willing to assist in any way she could throughout the PhD study.

Thanks are expressed to group members and colleagues in the Nottingham Geospatial Institute: Dr Yilin Xie, Dr Ruijie Xi, Dr Xiangdong An, Mr Rui Shang, Dr Fei Yang, Ms Jialin Xiao, Ms Chang Deng, Dr Hao Jing, Ms Roxanne Parnham, Dr Qusen Chen, Dr Denghui Wang, Dr Wang Gao, Mr Xu Chang, Mr Shuguang Wu, Mr Yijian Cui, Mr Xinao Wang, Dr Tung Dinh Nguyen, Mr Lei Zhao, Mr Kai Guo, Mr Brian Weaver. Thanks are extended to all the friends in the UK: Mrs Jun Ye, Ms Simin Wu, Ms Jinyi Li, Mrs Haiyan Xu, Mrs Sue, Mr George Ye and Mr Kairui Xie for their supports and help. Especially thanks to Dr Ruijie Xi, Dr Xiangdong An, Dr Fei Yang, Mr Rui Shang and Mrs Jialin Xiao for the unselfish knowledge sharing, academic discussions and supports. It is their kind help and support that have made my study and life in the UK an excellent time.

I would like to thank Prof Bijun Li and his team from Wuhan University for providing the CAVs testing devices.

Thanks are also given to friends Wanjun Li, Chang Yang, Tingting Zhang and Ronghui Tan for their continuous support during my PhD study.

Most importantly, thanks are given to my parents Lihua He and Baoguo He. Thanks for all the unconditional love and everlasting support for all these years. Thanks also to my grandma for the encouragement and support all the time. I would not have made it this far without their help.

Last but not least, the best achievement during my PhD study is finding my wonderful husband, Dr Lu Dong. He has been a true soul mate and the best listener during my good and bad times. I supposed that we both learned a lot of life, and strengthened our commitment and determination to each other as well. It is time to celebrate, you earned your PhD degree recently as well.

List of Publications

1. He Q, Meng X, Qu R. Towards a severity assessment method for potential cyber attacks to connected and autonomous vehicles[J]. Journal of advanced transportation, 2020.

2. He Q, Meng X, Qu R, et al. Machine Learning-Based Detection for Cyber Security Attacks on Connected and Autonomous Vehicles[J]. Mathematics, 2020, 8(8): 1311.

3. He Q, Meng X, Qu R. Survey on cyber security of CAV[C] 2017 Forum on Cooperative Positioning and Service (CPGPS). IEEE, 2017: 351-354.

Other Involved Publications during PhD:

4. Meng X, Roberts S, Cui Y, Gao Y, Chen Q, Xu C, He Q, Sharples S. Required navigation performance for connected and autonomous vehicles: where are we now and where are we going?[J]. Transportation planning and technology, 2018, 41(1): 104-118.

5. Xi R, Jiang W, Meng X, Zhou X, He Q. Rapid initialization method in real-time deformation monitoring of bridges with triple-frequency BDS and GPS measurements[J]. Advances in Space Research, 2018, 62(5): 976-989.

6. An X, Meng X, Jiang W, Chen H, He Q, Xi R. Global ionosphere estimation based on data fusion from multisource: Multi-GNSS, IRI model, and satellite altimetry[J]. Journal of Geophysical Research: Space Physics, 2019, 124(7): 6012-6028.

7. Xi R, He Q, Meng X. Bridge monitoring using multi-GNSS observations with high cutoff elevations: A case study[J]. Measurement, 2021, 168: 108303.

8. Xi R, Chen Q, Meng X, Jiang W, An X, He Q. A Multi-GNSS Differential Phase Kinematic Post-Processing Method[J]. Remote Sensing, 2020, 12(17): 2727.

9. Yang F, Meng X, Guo J, Shi J, An X, He Q, Zhou L. The Influence of Different Modelling Factors on Global Temperature and Pressure Models and Their Performance in Different Zenith Hydrostatic Delay (ZHD) Models[J]. Remote Sensing, 2020, 12(1): 35.

10. Xi R, Meng X, Jiang W, An X, He Q, Chen Q. A Refined SNR Based Stochastic Model to Reduce Site-Dependent Effects[J]. Remote Sensing, 2020, 12(3): 493.

List of Abbreviations

AV	Autonomous Vehicle
CAV	Connected and Autonomous Vehicle
CAVCS	Connected and Autonomous Vehicle Cyber Security
CAN	Controller Area Network
CFS	Correlation-based Feature Selection
CPU	central Processing Unit
CV	Connected Vehicle
DDT	Dynamic Driving Tasks
DoS	Denial of Service
DSRC	Dedicated Short Range Communications
DT	Decision Tree
ECU	Electronic Control Unit
ENISA	European Union Agency for Cybersecurity
ESA	European Space Agency
ETC	Electronic Toll Collection
FP	False Positive
FS	Feature Selection
GDPR	General Data Protection Regulation
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
ID	Identification
ITS	Intelligent Transportation System
LTE	Long Term Evolution
NB	Naive Bayes
NHTSA	National Highway Traffic Safety Administration
NIST	National Institute of Standards and Technology
OBD	On Board Diagnostics
R2L	Remote to Local
SAE	Society of Automotive Engineers
UML	Unified Modeling Language
U2R	User to Root
VANET	Vehicular Ad-hoc Network
V2C	Vehicle to Cloud
V2I	Vehicle to Infrastructure
V2V	Vehicle to Vehicle
V2X	Vehicle to Everything

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Connected and autonomous vehicles (CAVs), a subset of the Intelligent Transportation System, use hardware such as electronic control units (ECUs) and sensors, software such as entertainment systems and decision-making units, and data to conduct driving tasks with different levels of automation. Using these components, CAVs have the potential to not only drive without human assistance but also navigate, communicate with and react to their surroundings. The automation of CAVs is achieved by the installed sensors around the vehicle body that gather environment information to make decisions. Connectivity allows communication with vehicles, infrastructures and other road users, as well as guides navigation and vehicle reactions. The CAV is a combination of 'Connected Vehicles (CVs)' and 'Autonomous Vehicles (AVs)', forming the ability of CAVs to communicate with surroundings and conduct driving activities without human beings [1].

A variety of companies focus on the research and development of CAVs. One of the biggest Chinese Internet company Baidu released an open source autonomous driving platform named Apollo, which aims to address the challenging issues of precise sensing and decision making [2]. Tesla released its Autopilot for assistant driving and Summon system for assisted parking in 2015

and 2016, respectively [3]. An introduction to an enhanced Autopilot system, which could achieve autonomous driving in certain scenarios (on highways, for example), is found on Tesla's website [4]. As one of the biggest Internet companies in the world, Google is also a competitive player in the field of connected and autonomous driving. Google set up a company Waymo in 2009 to support the research and development of CAVs, and has already completed more than 2 million miles of road tests [5]. Another Internet company Uber, known for its taxi-hailing application, has also tested its own CAVs on public roads in the state of Arizona [6]. Except from Internet companies and electric vehicle companies, traditional vehicle manufactures such as Audi and Mercedes Benz have announced CAV initiatives as well. Audi has conducted 550km on-road test, based on its autonomous vehicle, "Jack" [7]. Mercedes Benz began to develop CAVs in the 1980s. Its latest S-class Benz vehicle has completed a 100km road trial in Germany [8].

To accelerate the development, governments also took actions to support by publishing relevant regulations and laws. In the US, regulations and laws on CAVs are established at the state level [9]. The Chinese government's ten-year plan, "Made in China 2025", proposes mastery of the key technologies of CAVs by 2025 [10]. In addition, the Chinese government has launched an abundance of CAV demonstration projects in China and established the Jiading district in Shanghai as the country's first public test area for CAVs [11].

Moreover, several CAV competitions sponsored by academic organizations, companies and governments have been held around the world. In the US, the DARPA Grand challenge was held in 2004 and DARPA Urban Challenge in 2007 [12]. In China, a competition focusing on the future challenges of intelligent vehicle competition has been held since 2008, sponsored by the National Natural Science Foundation of China [13]. Given the increasing number of research organizations participating, these competitions not only provide platforms for researchers to communicate but also raise public interest in CAV developments.

According to a survey conducted by the Boston Consulting Group, 55% of the public would like to try an autonomous vehicle or even buy one [14]. The majority of current CAV researches mainly focus on the functions of automation or connectivity. However, the CAV cyber security, a fundamental part of their development, is not sufficiently addressed. Cyber security is related to the functional safety of CAVs, which will have a direct influence on public trust and CAV commercialisation. According to the newly released UK CAV Cyber Security Principles [15], CAV cyber security should be considered at an early stage of CAV development (such as the design phase) and encompass the entire supply chain. This could prevent cyber security issues from arising in subsequent stages.

As probably the biggest mobile device people would use in the near future, CAVs however may cause severe consequences in people's lives, including not only private information leakage but also potentially fatal physical damages. Though the CAVs have not been commercialised yet, they have been involved in accidents several times and have already caused fatalities. In early 2018, an Uber autonomous vehicle hit a cyclist during road testing [16]. It is also reported that in the US [17] and China [18], Tesla vehicles have caused fatal incidents. Tesla announced that the driver's hands were not detected on the steering wheel for 6 seconds before the US accident. Although it was said that the Autopilot system was engaged, the Tesla vehicle should only be classified as a driver assistance system rather than a fully autonomous system according to the definitions of automation levels in Table 2.1.

Even more severe attacks "happened" to vehicles. In the US, white hat hackers have already attacked the Grand Cherokee successfully. They took control of the vehicle remotely from ten miles away and stopped it on a highway [19]. In 2019, the Tencent Keen Lab also announced that the researchers successfully took control of the Tesla S vehicles [20]. They used three dots installed on the road to mislead the testing vehicle and forced it to drive to the opposite lane. These accidents suggest that there may be severe and even life-threatening

consequences if CAVs are not designed and equipped with appropriate cyber security protection mechanisms. There is a pressing need to investigate CAV cyber security issues, even at this early stage in their development.

There has been little development of automation and communication-based technologies in traditional vehicles, and the issue of cyber security has often been neglected or considered less important. Although several mature standards and methods relating to cyber security exist in the field of computer science, they cannot be applied directly to CAVs. Compared to traditional networks, mobile networks, or traditional automobile networks, CAV cyber security has its own specific characteristics, which are shown in Table 1.1. Given these differences, the cyber security of CAVs should be considered specifically; cyber security strategies applicable to traditional networks or automobile networks cannot be employed directly.

Table 1.1: Comparison of CAVs with Traditional Vehicles/Mobile Networks

Compared to Traditional Vehicles	Compared to Computer Network/ Mobile Network
1. There are more ECUs and more code in the CAVs [21], which means more data to be processed. 2. There are multiple communication protocols in CAV, such as Controller Area Network (CAN) [22], 5G and DSRC [23], and therefore multiple data formats in the vehicles, requiring more processing time. 3. There are more connected functions, which means that the number of potential attack points is also increasing [25].	1. Besides information leakage, CAVs could cause physical damage or even fatal injuries. 2. CAV requires a higher detection rate as well as a shorter data processing time. In the Europe METIS project, latency is expected to be less than 5ms, and accuracy is expected to be 99.999 % when transmitting a 1600-bytes data package [24]. 3. The application scenarios are more complicated. CAVs are more likely to drive in unregulated areas such as parking lots, highways and rural areas.

1.2 Research Motivations

Due to these specific characteristics of CAVs outlined in Table 1.1, the research of CAV cyber security is of high importance and urgency. The UK published

the first CAV cyber security principles in the world [15]. The government, CAV companies and organizations are also looking for cyber security solutions. Innovate UK [26] has opened several Call for Proposals on CAV cyber security in recent years as have other organizations, such as the ESA (European Space Agency) [27].

The following considerations highlight the research significance of CAV cyber security:

1. The data and information transmitted by CAVs include not only communications within the vehicles (e.g., CAN) but also communications in V2V, V2I, V2C and everything in the CAV network. The specific characteristics of this complicated network must be captured and considered in a well-defined cyber security framework that can support future developments in cyber security protection. In addition, the framework should also guide the protection of newly-adopted technologies on CAVs.

2. The functions and commercialisation of CAVs have been researched for several years. It is believed that the CAVs could be ready for consumers in the 2020s [9]. Cyber security needs to be considered as soon as possible, and before the commercialisation, to protect the information and users' safety. However, currently there is no universal standard and regulation for CAV cyber security. The protection mechanism could not be guaranteed. The lack of relevant security data also poses challenge for further studies on autonomous detection on attacks to CAVs.

3. Secure communication is the foundation for the successful development of CAVs, to avoid both private information leakage and physical damage to properties or people. The CAV cyber security needs to be emphasised across different sectors and throughout the entire supply chain of the automobile industry.

1.3 Research Aims and Objectives

The aim of this thesis is to provide a new Anomaly Detection Framework that will enhance CAV cyber security through the adoption of machine-learning methods for CAVs by defining and assessing attacks, collecting data from simulated and real world environments, adapting machine learning models, and improving the performance by feature selection methods.

The Objectives of this research are as follows:

1. Critically review existing cyber security frameworks and mechanisms for use in CAVs. Because CAV cyber security is a new and emerging research field, other relevant literature, including cyber security of traditional networks, traditional automobiles, and Vehicular Ad hoc Network (VANET), will also be reviewed to support the interdisciplinary research of CAVs. The gaps and limitations of current research on CAV cyber security will then be identified for further in-depth analysis and investigations.

2. Define the terminology related to CAV cyber security and assess potential attacks on CAVs by specific CAVs cyber security severity assessment method. As there is no general definition of CAV cyber security, the research domain must be properly defined. In addition, cyber security research must identify and define the types of attacks that could occur. A new universal CAV cyber security attack severity assessment method will be proposed, which could help to understand the priority for protecting against and responding to attacks.

3. Collect new CAV cyber security data sets within the new CAV framework. Because CAV cyber security data sets are lacking in the existing literature, new CAV cyber security data sets will be collected from both simulated and real world environment. These data sets will cover different attack types to CAVs, and help to build machine learning models as well as evaluate them in real world usage.

4. Develop and evaluate machine learning models for CAVs based on new CAV cyber security data sets. Machine learning models will be adapted to each of

the data set to assess their performance on detecting different cyber attacks to CAVs in various situations. The performance of the accuracy, false positive (FP) rate and runtime will then be compared, the most appropriate model will be suggested.

5. Improve the performance of the models by feature selection methods. Feature selection methods are to be investigated to enhance accuracy, lower the FP rate, and decrease detection time of machine learning models on CAV attack detection. The selected features towards different attacks will be investigated, which could help to understand the importance of each feature.

1.4 Research Methods and Procedures

The focus of this thesis is to develop a framework for the cyber security of CAVs. Within the new framework, potential attacks could be defined and machine learning methods could then be used to detect them, thus enhancing CAV cyber security. The proposed solution basically follows the cyber security framework introduced by the NIST (National Institute of Standards and Technology) [28]. The NIST framework core consists of five main functions: identify, protect, detect, respond and recover. However, the framework is designed for companies or organizations to conduct commercial risk assessments to reduce cyber security risks to an acceptable level. For CAVs, the framework is adapted and changed based on specific characteristics of CAVs outlined in Table 1.1.

This work acknowledges that cyber security attacks on CAVs may cause severe physical damage. Thus, after identifying the potential cyber attacks on CAVs, a new severity assessment method needs to be proposed to consider more CAV specific factors.

Once potential cyber attacks are identified, the detection of attack represents the next important key issue. Machine learning tools have been identified to be highly successful and powerful in the recognition of patterns in data.

This research generates and labels new CAV data sets, including in-vehicle and inter-vehicle data, to train machine learning models for CAV anomaly detection. The trained machine learning models that recognized the patterns between data could be adapted to new data sets to detect the anomalies.

This work also acknowledges the demand for high accuracy and low latency in CAVs responses to real world driving situations. These are often dynamic and unpredictable. After the construction of the machine learning models, feature selection methods are used to reduce the number of attributes in the data set, thus to reduce the runtime without affecting the accuracy of the model. With the attributes selected through the feature selection process, the models are shown to be well adapted to achieve better performance. The models are also adapted to a real world data set to evaluate their applicability in new and complex scenarios.

The overall structure and methodologies of the thesis are outlined in Figure 1.1. A more detailed description will be given in Section 1.6.

1.5 Contributions of the Research

The contributions of this research are listed below. The main contribution of the thesis is to build a new machine learning-based anomaly detection framework for CAV cyber security, which incorporates severity assessment, data collection, attacks detection and performance improvement. The framework has been adapted and validated in the simulated and real world environment, showing it to be effective for assessing and detecting CAV cyber attacks. It provides guidance and a baseline for further CAV cyber security researches.

a. A new proposed severity assessment method specific to CAV cyber security. The new set of terminology of CAV cyber security, including CAV cyber security, CAV network, and CAV cyber attacks, is defined to build the theoretical foundation of the CAV cyber security framework. The priority of each attack could then be understood and the most severe attacks could be resolved with

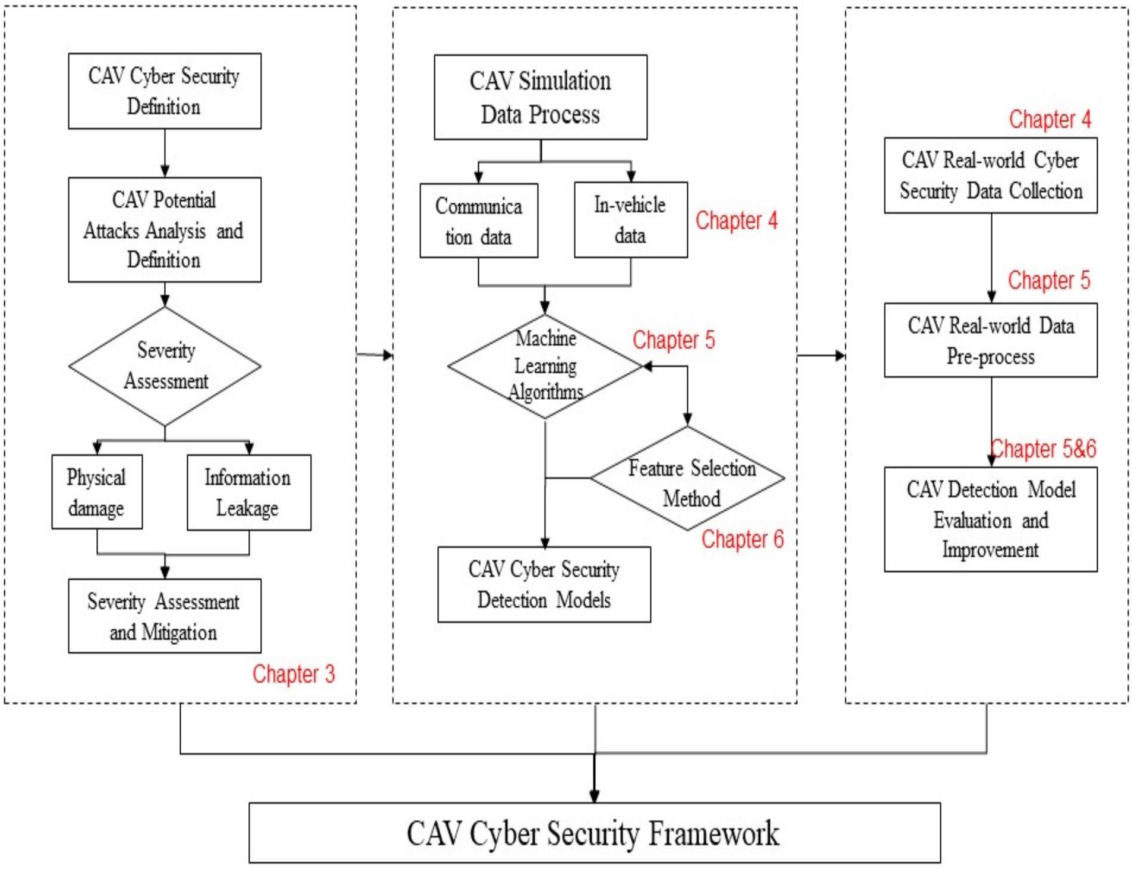


Figure 1.1: *Thesis Structure and Methodology*

a higher priority by adapting the proposed assessment method. Besides, this newly proposed assessment method is also extendable in the fast evolving area of CAV research and development. This contribution provides guidance for CAV cyber security severity assess, which could help researchers understand the attacks and then take corresponding reactions towards them.

b. Four new CAV cyber security data sets covering potential attacks in simulated and real world environment. CAV-KDD data set generated from cyber security benchmark KDD99 covers potential attacks to inter-vehicle communications; Simulated Simu-CAN data set and real world KCAN-CAV data set cover DoS attacks and Fuzzy attacks on in-vehicle communication environments; Self-collected CAV-RW data set addresses the limitations of these three data sets, restoring the DoS and Fuzzy attacks scenarios to the greatest extent in the real world. These data sets help build and evaluate machine learning models towards different attacks, filling the research gap of lacking relevant

data sets in the field of CAV cyber security. Other researchers could also use these data sets to conduct relevant research and compare the results.

c. Machine learning models, Decision Tree and Naive Bayes, to detect the CAV cyber attacks on the new data sets. The machine learning models are built and evaluated on the data sets to assess the performance of accuracy, FP (False Positive) rate and runtime. It is found that machine learning is effective to detect CAV cyber attacks. The comparison results show that the performance of Decision Tree model is superior than that of Naive Bayes. The possibility of using machine learning models is evaluated in the thesis, based on which other researchers could use different machine learning algorithms to compare and improve the results.

d. Improved performance of machine learning models by feature selection methods. The feature selection methods provide a possible solution for dynamic driving environments. In data set with enough number of attributes, the feature selection methods are considered to be effective. Besides, feature selection methods also help to prioritise the importance of attributes, by which only crucial attributes are collected if data storage and computation power are limited. It is found that time frequency and data field content are the most important attributes in detecting DoS and Fuzzy attack, respectively. The best performance model and important attributes towards different attacks are also suggested, which could provide a baseline for further research. In addition, the important attributes could also provide guidance for CAVs data collection in future research.

1.6 Thesis Overview and Structure

The rest of the thesis is organised as follows.

Chapter 2 presents an overview of researches on CAVs as well as CAV cyber security. The existing research emphasises that cyber security is a fundamental part of the deployment of CAVs. Components of building cyber security

framework are also reviewed from the aspects of risk assessment, related data sets, anomaly detection and feature selection. The research gaps with respect to existing research are highlighted in this chapter.

Chapter 3 describes the preliminary development of the proposed CAV cyber security framework. In this chapter, the terminology of CAV cyber security is firstly defined followed by the potential attacks on CAVs, which are defined, analysed and categorised from the perspective of automation and connectivity. A new CAV cyber security severity assessment method is proposed, in which a new set of severity assessment criteria is provided to support the categorisation and priority of potential attacks. Based on the criteria, the most severe attacks on CAVs are identified, and corresponding mitigation methods are suggested to resolve them.

Chapter 4 introduces all the data sets newly collected and analysed in the thesis. These four different data sets cover most severe attacks defined in Chapter 3 in simulated and real world environments. The newly retrieved data set CAV-KDD is generated from the traditional network benchmark data set KDD99 by removing redundant data and irrelevant attributes. The Simu-CAN data set is simulated using a CAN tool, and attack scenarios are simulated to inject the attack data into the data set. Real world data sets, including KCAN-CAV data set of inter-vehicle communication, and self-collected CAV-RW data set from a real CAV, are used to validate the performance of machine learning models in the real world environment. The data sets in this chapter support the follow-up machine learning research in the thesis.

Based on these data sets, two machine learning models, namely Decision Tree and Naive Bayes, are built and evaluated in Chapter 5. The performance metrics of accuracy, FP rates and runtime are then discussed and compared. This chapter demonstrates the applicability of machine learning models to detect the anomalies on CAV cyber security with a concern of further improvement on model's performance.

In Chapter 6, feature selection methods based on the specific characteristics

of CAVs (i.e., the highly dynamic environment and low fault tolerance) are used to improve the performance of machine learning models. The accuracy, FP rate and detection time of each data set are compared with where feature selection methods are not applied, in order to demonstrate any improvements gained through the use of the methods. In addition, the best performance model and important attributes are also suggested to provide guidance for further research.

Chapter 7 presents the conclusion of the thesis. The limitations of the research, the challenge ahead of CAV cyber security and future work are also discussed.

Chapter 2

Literature Review

2.1 Overview

The automobile industry has developed for several centuries, and the number of ECUs (Electronic Control Units) installed in vehicles has also been increasing in the past few years. The development of CAVs is becoming a major priority of the automobile industry. They have spent a huge amount of money on their development, and all car manufactures are competing for the leading position in the area of CAVs [29]. Nowadays, an increasing number of sensors and applications are used on vehicles to build a more efficient and reliable driving environment. It is said that one modern vehicle now could have more than 100 million lines of code [30] to support its driving functions.

Some initial applications of CAVs have already been installed on modern vehicles. For example, the ADAS (Advanced Driving Assistant System) has been installed and used in many commercialised vehicles [31]. Vehicle platooning technology is also used in the lorries [32]. Besides the applications on vehicles, the whole driving environment, such as infrastructures, is becoming more intelligent as well. With wireless communication technology usage, such as RFID (Radio Frequency Identification) [33], passing vehicles could be charged automatically at parking lot entrances and toll stations on highways without stopping, improving traffic efficiency greatly. Traffic control systems such as

traffic light control system [34] to avoid congestion are being used in cities around the world as well. The concept of VANET (Vehicular Ad hoc Network), which is a crucial part of the intelligent transportation system, is also adapted into real world usages in certain scenarios to increase the efficiency of the whole driving system [35].

All of these technologies try to improve the automation level and connection degree of modern vehicles, and help the vehicles make driving decisions independently without human. However, more vulnerabilities will be exposed on CAVs with the installation of new ECUs and applications as the number of ports to outsides is increasing as well. With wireless communication, all the vehicles in the communication range could become the attack targets [25], which is even worse than wired communications.

This chapter introduces the background of CAVs and existing studies on CAV cyber security. It also reviews the relevant literature on potential attacks on CAVs and the mitigation methods that could be used against these attacks. The current CAVs standards and industry reports are also reviewed in this chapter. Finally, relevant approaches and methods of building a cyber security framework, including risk assessment, data sets, anomaly detection and feature selection, are reviewed in this chapter as well.

2.2 Connected and Autonomous Vehicles

The SAE J3016 Standard is a de facto standard for autonomous driving nowadays. The category of automation defined by J3016 has been widely used. The SAE International defined "driving automation" as that the system could conduct part or all DDT (Dynamic Driving Tasks) continuously [36]. DDT is defined as three different levels by the SAE J3061 standard, namely operational functions, tactical functions and strategic functions. The relations of these three functions are illustrated in Figure 2.1 [36]. Operational functions contain basic motion control, including lateral and longitudinal motion

controls. Tactical functions contain all the operational functions plus OEDR (Object and Event Detection and Response). In current DDT performance, the strategic functions, including destination and waypoint planning, are not included.

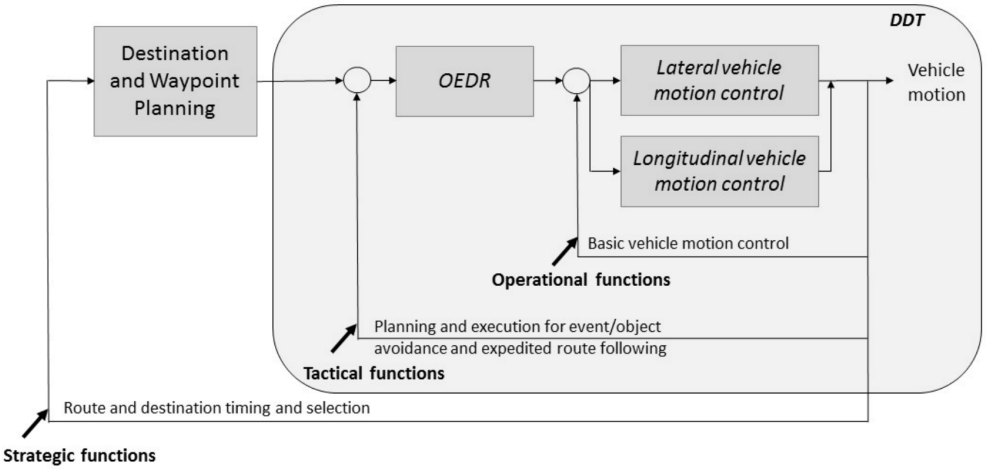


Figure 2.1: *Schematic View of Driving Tasks Showing DDT Portion*

The response by either users or the system to perform DDT when a system failure happens is defined as DDT fallback by SAE International. ODD (Operational Design Domain) is considered as the driving system requiring a specific running environment, including environmental, geographical or time restrictions. For example, some autonomous driving vehicles only operate or test in a closed environment [37], which indicates that the vehicle is still designed under a limited ODD. Based on the DDT performance, DDT fallback and ODD, SAE International then defines the vehicle automation into 6 different levels, which are shown in Table 2.1 [36].

The definition of CAVs varies in different descriptions. In the UK, the government set up a government centre called the "Centre for Connected and Autonomous Vehicles" in 2015 [38]. This centre published a report on Connected and autonomous vehicle research and development projects in 2018 [39]. The House of Lords also published the report "Connected and Autonomous Vehicles: The future" in 2017 [40]. Other organizations, including the British

Table 2.1: SAE Automation Levels

Level	Name	DDT		DDT Fallback	ODD
		Sustained Motion Control	OEDR		
0	No Driving Automation	Driver	Driver	Driver	N/A
1	Driver Assistance	Driver and System	Driver	Driver	Limited
2	Partial Driving Automation	System	Driver	Driver	Limited
3	Conditional Driving Automation	System	System	Fallback-ready user (becomes the driver during fallback)	Limited
4	High Driving Automation	System	System	System	Limited
5	Full Driving Automation	System	System	System	Unlimited

Standard Institution (BSI) in the UK, also published a standards strategy report on CAVs in 2017 [41].

Some publications have also used the name of Connected and Automated Vehicles. For example, the Transport Systems Catapult [42], an innovation centre in the UK, used the term Automated on its website. As a rapidly developing subject, the naming of CAVs is not consistent in the literature, at present. The thesis therefore use the term 'Connected and Autonomous Vehicle', which is the same as 'Connected and Automated Vehicles' in the literature.

CAVs are attributed with the features of wireless connectivity and automation. Connected means that the vehicles rely on data sent from other vehicles or infrastructure to plan their routes and communicate with other surrounding vehicles within a connected network. Full automation means that these vehicles can comprehensively conduct dynamic driving tasks and recovery actions automatically, in real-time, without driver's intervention [43]. In [44], the authors concluded that a modern autonomous vehicle normally contains three crucial elements, which are sensors, on-board computers and actuators.

These elements could help the vehicle to understand surroundings and make corresponding reactions.

CAVs could bring many benefits to current transportation and people's daily life. First of all, traffic safety could be enhanced by CAVs. It is reported that 90% of traffic accidents are caused by drivers' fault, which means that by using CAVs, the same proportion of accidents could be avoided [9]. Secondly, traffic efficiency could be significantly improved by CAVs. Using CAVs could increase the road capacity, because the car spacing could be shortened when driving [45]. Thirdly, fuel consumption could also be reduced, and the air pollution caused by vehicles could be controlled. The car-sharing program could also be expanded. It is studied that one sharing CAV could replace eleven traditional vehicles on car-sharing rental programs, which could help to reduce the emissions [46]. In [47], the authors conducted a comprehensive research about the implications that may be caused by CAVs, including fuel consumption, travel choices, public health, etc. With the increasing of automation, the impacts would also be multiplied.

Though the development of CAVs could bring lots of benefits to our daily life, it could also cause some problems such as reliability or safety. These unintended concerns may reduce the customers' acceptance to CAVs. Many researches emphasised the possible consequences on safety. In [48], the authors said that though CAVs could reduce human errors, the machine faults still cannot be avoided. Several fatal accidents around the world indicated the unreliability of CAVs. In [49], the authors found out that if CAVs and traditional vehicles were mixed on the road, the number of conflicts would even increase in certain areas such as the intersections or low car spacing roads. The transition period from traditional vehicles to CAVs would be dangerous. In [50], the author said that though safety is crucial to CAVs, functional safety has already been inappropriate for CAVs as security also needs to be considered.

It could be seen that all the researches agreed that the development of CAVs is promising and will be beneficial to several aspects in our daily life and the

whole society. However, there still exists several challenges ahead to overcome, in which CAVs cyber security needs to be considered first as it could impact both safety and security.

CAV cyber security is an important issue in the development of CAVs, and also a big challenge. In the next section, CAV cyber security will be reviewed and discussed.

2.3 Cyber Security Research on CAVs

Though the technical research of CAV cyber security is still developing and no mature technologies has been applied to CAVs yet, the cyber security are still accounted of by governments around the world. Several regulations, laws and best practices have been published by governments.

In the US, the NHTSA (National Highway Traffic Safety Administration) published best practice of modern vehicles in 2016 [51]. One year later, in 2017, the second version of the safety report was also published [52], which was named "a vision for safety". Both of the reports tried to define and regulate relevant issues on vehicle cyber security. However, they are not specific to CAVs. The best practice focused on modern vehicles and the safety report only focused on automated driving. Another regulation called the "Spy Car Act", which was published by the US government in 2015, also aims to protect the privacy and security of self-driving vehicles [53]. At the state level, several states, including California and Massachusetts, introduced or updated new and existing laws to ensure cyber security for CAVs [54]. Other states such as Georgia and Michigan also passed new laws to ensure cyber security in general rather than specific to CAVs [54].

ENISA (European Union Agency for Cybersecurity) in the EU also published relevant researches on CAV cyber security. In 2017, a report on cyber security and resilience was published [55], listing all the vulnerable parts of smart vehicles and possible threats to them. In addition, the GDPR (General Data

Protection Regulation), which took effects from 25 May 2018, was also an advanced approach for privacy protection in Europe [56]. In 2019, a new good practice for security was published to improve vehicles' safety [57]. Both of these two documents are guidelines and suggestions, but not compulsory. It is still a signal that the EU paid attention to the cyber security of CAVs. The GDPR regulates the use of personal data in the EU, which is a model for the data protection law for other areas. As CAVs would store highly sensitive personal data such as the usual locations or bank account information for payments, GDPR will be definitely used in future CAVs.

In the UK, new CAV cyber security principles had been proposed [15], which made the UK the first country to consider CAV cyber security at the national level. However, these principles are only guidelines, which are very general. In Asia, China and Japan also updated the existing laws to secure cyber security and privacy. In China, the Cyber Security Law of the People's Republic of China had enacted in 2017, which requires the enhancement of cyber security. In late 2017, China has introduced another guideline for developing Intelligent and Connected Vehicle, which emphasised the importance of CAV cyber security [58]. The guideline clearly pointed out that cyber security standards of CAVs, including the technical and privacy requirements, need to be built. In Japan, the Personal Information Protection Commission would monitor and protect the personal information [54].

It could be seen that the governments have realised the importance of CAV cyber security. However, as the technology is not mature, governments still do not force automakers and OEM suppliers to take compulsory actions. But, initial attempts about CAV cyber security need to be considered carefully, and protection mechanisms need to be proposed to ensure not only security but also safety on CAVs.

Researchers have emphasised that the protection of cyber security is the most important requirements of users to use CAVs [59]. There are various attempts to discuss CAV cyber security. In [60], the authors discussed the possible cyber

security attacks on autonomous vehicles. After listing all the possible attacks, the authors then gave mitigation solutions to each attack. It is recommended that it is important to keep sufficient redundancy in autonomous vehicles. Sufficient sensors data could help vehicles to know the surroundings and positions. They believed that GNSS spoofing and fake message injection are the most threatening risks among all these attacks. Both of these attacks will put threats to passengers' lives. They believed that anti-spoofing hardware and authentication methods are needed in autonomous vehicles.

In [61], the authors divided the attacks to vehicles into two classifications. One is the attack to the entertainment system on the vehicles, including the audio system or the mobile applications installed on the vehicles. The other classification is the attack to the CAN, which is even more dangerous to the vehicles. Unlike the computer DoS attack focusing on communications, the DoS attacks to vehicles could be different. For example, if the attackers attack the heating function of the driver's seat, the power of the vehicles would be consumed quickly at a short time. This could be extremely dangerous, especially to electric vehicles.

In [62], the authors discussed cyber security in connected vehicles. The authors believed that the vehicles would be more vulnerable with the increasing connectivity. This paper described the possible attack scenarios, including USB update attacks, communication attacks and malicious application installation. A system using machine learning methods was then built to detect the anomaly behaviours in CAN-Bus (Controller Area Network) and the operating system.

In [63], the authors attempted to use the categories of cyber security in computer science to describe the possible attacks in CAV. The possible attacks were divided into passive attacks and active attacks. The passive attack is easy to prevent but difficult to detect, while the active attack is easy to detect but difficult to prevent. In the thesis, mitigation methods were recommended, including authentication and encryption.

In [64], the authors pointed out that the current vehicle safety standard ISO26262 did not consider the security issues to avoid both unintentional and intentional attacks. Currently, there is no existing universal security or safety standard for CAVs. Therefore, a systematic definition of attacks and attack analysis methods is highly desirable for the development of CAVs.

In [65], the authors assumed that connected vehicles were similar to all the Internet devices, and cyber security should be considered as a fundamental part of connected vehicles development. The authors then discussed the potential cyber attacks on V2I (Vehicle to Infrastructure) communication and proposed a novel cyber security architecture called CVGuard to detect the attacks in V2I. In this paper, the CVGuard reduced 60% DDoS (Distributed Denial of Service) attacks created vehicle conflicts.

In [25], the researchers pointed out that modern cars were already new targets for hackers. Engines, doors and brakes could all be possible vulnerable points. In addition, nowadays, the attackers did not need to approach the target vehicle physically. All the vehicles in the communication range could be hacked. The authors also listed OBD (On-Board Diagnostics) threat, DSRC communication, Malware and automobile apps as the most vulnerable parts on vehicles. The authors also offered solutions to address the cyber security issues, including OTA (Over-the-Air Technology) solution, cloud-based solution and layer-based solution.

Real attacks have also happened several times. White hackers from Tencent Keen Lab have successfully attacked Tesla [66]. The attack was conducted in September 2016. A remote attack to Tesla Model S had successfully conducted both in driving and parking modes. The white hackers accessed the Tesla vehicle from Wi-Fi, and modified the CAN message successfully. After this attack, Tesla published an OTA update immediately to fix the problem.

Three years after this attack, in 2019, the Keen lab conducted another successful attack to disrupt the rain wipers of Tesla [20]. The lane recognition function was also hacked by sticking three white round dot stickers on the

road only. With wrong lane recognition, the vehicle drove to the opposite lane. In addition, the white hackers also showed that the steering wheel could be controlled by a gamepad, even when the Autopilot was not running.

Because CAV is a newly-developed research topic, the amount of cyber security research on CAV is limited. Except for the cyber security in CAV, there is also research on the VANET (Vehicular Ad hoc Networks), which shares a lot in common with the connected functions of CAVs.

VANET uses V2V communication and V2I communication to help vehicles gathering traffic information [67], while CAVs extend the boundaries to V2X communication. VANET is a mobile ad hoc network, where the vehicles are the mobile nodes [68]. In [68], the authors listed the possible privacy and security challenges to the safety of VANET, including the attacks on confidentiality, integrity or data trust. They claimed that encryption is important to VANET. In [69], the authors concluded that the VANET has three specific characteristics: frequent vehicle movement, time-critical response, and hybrid architecture. Other listed attacks include bogus information, DoS attacks, Masquerade, GPS spoofing, etc. The authors also proposed several mitigation methods, including public key, certificate revocation approaches and ID-based cryptography.

As it could be seen from the literature listed above, the majority of the researchers believed that cyber security is a fundamental part of the development of CAV, which needs more research and investigations. On the other hand, most authors agreed that increasing connected and automated functions will increase the possibilities of cyber attacks. However, these papers only discussed cyber attacks on one aspect, either on connected functions or autonomous functions. The in-vehicle cyber security and inter-vehicle cyber security were discussed separately. There were attempts to discuss the most severe attacks, but there were no systematic evaluation criteria. Without clear definition of CAV cyber security and supportive data set, the detection of cyber attacks towards CAVs could not be conducted. Meanwhile, the literature

on CAV cyber security was limited, and this topic needs more investigation and research efforts. Awareness of cyber security in CAV should also be raised.

2.4 Components of Cyber Security Framework

Relevant researches on building a cyber security framework were reviewed from four aspects: Firstly, the risk assessment, which could help to enhance the CAV cyber security framework after definitions of attacks. The risk assessment could help to understand and prioritize the potential attacks. Then, as the indispensable part of the detection, relevant cyber security data sets were reviewed. The methods of intrusion detection were also reviewed to discuss the research gaps in the detection of CAV cyber security attacks. Finally, feature selection methods were reviewed, aiming to improve the performance of intrusion detection models.

2.4.1 Risk Assessment

Risk assessment is the initial step in a proposed cyber security framework because it requires the definitions of all the potential attacks to the target. Only by knowing and understanding potential attacks could corresponding detection mechanisms and corresponding reactions be taken. Meanwhile, the severity of each risk would also be evaluated. Some researches discussed the risk assessment of potential attacks to CAVs or vehicles. The attempts of these researches helped to identify the attacks and severity of different risks systematically. However, as the number of relevant researches of risk assessment in CAVs is limited, the risk assessment from information security or traditional vehicles have also been investigated to gain a comprehensive understanding of risk assessment.

In [70], the researchers discussed existing risk assessment on information security in the field of computer science. The researchers pointed out that the main aim of risk assessment is to investigate all the potential risks to the

system. After that, all the risks should be evaluated carefully so that the mitigation methods could be proposed appropriately. This paper also defined the differences between risk, threat and vulnerabilities, because normally these definitions are misused in risk assessment in the entire cyber security framework. In this study, the researchers defined that the potential attack targets, or the assets, had vulnerabilities. The attackers could make use of the vulnerabilities, which will pose threats to the system. The possible relationship between vulnerabilities and threats was then called risks.

In [71], the authors built a risk assessment framework for smart vehicles. The smart vehicle is the initial concept of the CAV. The authors tried to evaluate the importance of different assets. The evaluation was from three aspects: safety, privacy, and operational. After analysing the importance of assets, the threat or the vulnerabilities would be calculated based on different severity levels. The authors assigned a numeric value to each threat and vulnerability. The risk was then calculated based on it. The numeric value could help evaluate the risk and prioritise different risk, by which mitigation methods could be proposed. In addition, as it is impossible to conduct all the potential attacks on smart vehicles, it is also crucial to calculate the risk before the developments. However, the framework built in this paper only discussed the evaluation of risks. It did not discuss the potential attacks on smart vehicles.

In [72], the authors proposed a systematic way to analyse the risk to the vehicle IT system. The paper introduced a well-adapted equation that risk equals the probability of an accident and the possible losses through the accident in the field of engineering. The authors then defined the possible losses into three different categories: safety, financial, and operational. Damage to these three aspects could cause risks to the vehicle. In the paper, the authors also ranked the damages from three aspects into 4 levels. Each level has its own factor from 0 to 1000, which helped to understand the severity of each risk. The paper also pointed out that in the current automotive standards, such as the ISO26262, risk assessment was not enforceable, and there was no universal

standard to set the baseline of risks.

In [73], the authors introduced the threat analysis and risk assessment in traditional vehicle cyber security. In the paper, the severity was based on safety-related components and privacy-related components. Each component has five different levels, from S0 to S4. The authors also emphasised that though there were standards to assess the hazard and risks in vehicles, the cyber security issues were not considered carefully. Thus, privacy and safety should be considered at the same time on the developments. Meanwhile, it should also be noted that in the paper, the definitions of threat and risk were not clearly classified.

In [74], the authors tried to propose a novel method to conduct the hazard analysis and risk assessment for CAVs on Level 4, which means that the vehicle could perform the driving tasks without people in a controlled environment. The new assessment method was designed to protect the functional safety of CAVs. The authors defined relevant terminology, including hazard, hazardous events, to CAVs, as current terminology is all designed for traditional vehicles. Then, the safety goals of different operational functions were set. The authors also emphasised that the terminology definition is an important part of risk assessment. However, the paper only discussed functional safety, which is more related to the automation functions. The connected functions were not considered in this paper.

There were also some published patents for the risk assessment for CAVs. In [75], the researchers invented a new method to assess CAVs risk by collecting and storing sensor data and operational data. These two types of data were then combined and compared so that the driving environment could be simulated. Based on the reproduced driving events, the risk of loss of the CAVs could be evaluated. In [76], the researchers proposed a new method to assess the vehicle's risk based on the location. This method is especially useful when there is a financial transaction, such as the car renting situation. The method could help to check the safety level of the current location of vehicles and

evaluate the transaction risks.

Based on the above risk assessment researches, several conclusions could be drawn.

First of all, all the researchers agreed that terminology definition is a crucial part of the risk assessment, which is also an irreplaceable part of the CAV cyber security framework as currently there is no universal definition of CAV cyber security terminology. Even in some studies, the terminology was misused. For example, smart vehicles, CAVs, and automated vehicles were used interchangeably in several descriptions. However, the concepts are different distinctly. Only CAVs focus on both automation and connection at the same time.

Secondly, though there were risk assessments for CAVs, it was still mainly on the automation functions, indicating that only functional or operational risks were considered. While in CAVs, the connected functions would also bring new risks to the drivers and users. In order to build a more comprehensive framework, both connected and automated functions of the vehicles need to be carefully considered.

Thirdly, the current risk assessment method was based on the experience of experts or researchers. The severity of risks could not be calculated systematically. In addition, the potential risks and attacks to CAVs were not fully investigated as well. Though there were studies related to the systematic risk assessment or the potential attack investigation, the studies did not combine the two parts.

Based on the above conclusions, it could be found that to build a CAV cyber security framework, the risk assessment is crucial, which will establish the solid foundation of the whole framework. Only by defining attacks and assessing the risks can the cyber security framework be systematic and reliable.

2.4.2 Related Data Sets

Being the most fundamental part of building a robust attack detection model and completing a comprehensive CAV cyber security framework, the experiments and tests on CAVs are essential. However, considering the special characteristics of CAVs, especially the potentially life-threatening consequences of CAV cyber security, it is impossible to conduct the attack detection directly on CAVs without first collecting attack data sets. To conduct the attack detection on CAVs, attack data sets need to be collected first. Some researchers have discussed attack detection, and they have used various attack data sets. In [77], the authors reviewed 65 papers related to CAV cyber security based on three main categories: threat, solution, and research. From their review, they found that 37 of the studies used simulated data sets. However, the in-vehicle data sets and vehicle communication data sets were investigated separately rather than combined. In addition, the safety data of the CAVs was missing. Most of the studies only focused on the connected data. It should be noted that, though in this research, 37 pieces of research used different data sets to perform cyber security research, all the data sets are not open source. It is not easy to obtain the data sets directly, and it is difficult to obtain permission to use them in this research.

Car manufactures, car hackers and internet companies also conducted attacks on real world vehicles or CAVs, as mentioned in Section 2.3. Vehicle attack data sets must be collected during the attacks, but most of the data sets could be extremely difficult to obtain due to safety or commercial issues.

In [78], the Hacking and Countermeasure Research Lab in Korean published the car hacking data sets on its website. The OTIDS (CAN Data set for intrusion detection) contained three different attack types, which are DoS attack, Fuzzy attack and Impersonation attack. Besides the attacks data, one attack free data set was also provided in the data sets. The total amount of attack data and normal data are beyond 5 million. It should also be noted that these

data sets were collected from a traditional vehicle, so they are lack of some certain data types. Though the data types and attack types are limited, it is still a good attempt to evaluate the models in data sets collected from real world. These data sets will be introduced and analysed in detail in Chapter 4. Data is a necessary part of the attack detection of CAV when using machine learning algorithms. However, the lack of CAV cyber security data presents a research gap for current CAV cyber security research. Except for the data sets mentioned above, current CAV data sets mainly focus on the maps or sensor data collected by CAVs. There are some well-known data sets provided by different research organizations or companies, which are listed in Table 2.2.

It could be found from the table that almost all the data sets focused on the autonomous functions of CAVs, including the functions and sensor data. However, the connected functions of CAVs are important as well. Since there is no universal standard for the attacks to the communications of CAVs currently, the related data sets of attacks to traditional network communication are also reviewed. Several benchmarks of network attack detection are introduced below.

1. KDD99 data set:

KDD99 is the most well-known benchmark for intrusion detection [92]. It was first published in the Third International Knowledge Discovery and Data Mining contest, the aim of which is to build an intrusion detection system. Though the KDD99 data set has been published for more than 20 years, it is still used in current researches. The training set has 4,898,431 instances and the testing set has 2,984,154 instances [93]. It also provided a 10% data set with the same proportions as the original data set. 39 different attacks belong to 4 main types: DoS (Denial of Service), R2L (Remote to Local), U2R (User to root), and probing. Each data in KDD99 has 41 attributes. A more detailed description of KDD99 can be found in Chapter 4.

2. NSL-KDD

The NSL-KDD data set evolved from the KDD99 data set, which was first

Table 2.2: Current CAVs-related Data Sets

Data Set	Provider	Data Type
KITTI [79]	Karlsruhe Institute of Technology; Tyota Technological Institute at Chicago	Stereo sequences data; 3D point clouds; 3D GPS/IMU data; Calibration; 3D object labels
Audi Autonomous Driving Dataset (A2D2) [80]	Audi	2D semantic segmentation; 3D point clouds; 3D bounding boxes; Vehicle bus data
Waymo Open [81]	Google	3D LiDAR point clouds; 2D camera images; urban and suburban area 2s sequences data
Apollo [82]	Baidu	Trajectory dataset; 3D Perception Lidar Object Detection and Tracking dataset
Lyft [83]	Lyft	Raw Lidar and camera inpus
Ford Autonomous Vehicle Dataset [84]	Ford	3D maps; 3D Lidar point clouds; Calibration
BDD100K (Berkeley Deep Drive Dataset) [85]	Berkeley	Images and videos
Mapillary Vistas Dataset [86]	Mapillary Research	images; object categories; different time/season/weather data
Cityscapes Dataset [87]	Cityscapes Team	Images; GPS Coordinates; Ego-motion data; Temperature data; Different cities/time/season/weather data
nuScenes [88]	Motional (Aptiv's expertise in automotive technology and Hyundai Motor Group)	Camera images; Lidar and Radar data; Object detection bounding box
CamVid [89]	the University of Cambridge	Video sequences; Labelled images;
H3D (Honda 3D Dataset) [90]	Honda	360 degree Lidar point cloud; interactive traffic scenes; 3D bounding box labels; Traffic participants
Oxford Robotcard Dataset [91]	Robots	Camera images; Lidar, GPS, INS ground truth; Different time/season/weather data

published in 2009 [94]. Some drawbacks still exist in the KDD99 data set. The NSL-KDD overcame the limitations and deleted all the redundant data in KDD99, so the data frequency could not affect the prediction results. As in KDD99, the NSL-KDD also contains 39 different attacks. The data in the NSL-KDD contains 41 attributes.

3. UNSW-NB15

Compared with KDD99 and NSL-KDD, UNSW-NB15 is the latest intrusion detection data set, which was published in 2015 [94]. Unlike the KDD99 data set, this set has 49 features and 9 different attacks, including Reconnaissance, shellcode, exploit, fuzzers, worm, DoS, Backdoor, Analysis, and Generic [94]. Because KDD99 and NSL-KDD data sets have limitations such as data bias, a research group from the Australian Center for Cyber Security used a tool called IXIA to generate this new data set [95]. Compared with other data sets, this data set has more attacks representing low footprint attacks [92]. It should also be noted that in KDD99, there are 14 unseen attacks in the testing set, while in UNSW-NB15, the attack types are the same in both the training and testing sets [94].

After reviewing all the relevant data sets, it could be found that there is a lack of real CAVCS data, which is due to the confidentiality of the organizations or car manufactures as well as the difficulties in conducting cyber attacks on CAVs. Collecting cyber security data on CAVs is an irreplaceable step to conduct further studies. In addition, as there are no universal CAV communication standards, communication attack data for CAVs is also difficult to obtain. Though there exist network attack data sets, the data cannot be used without processing due to the differences between CAVs and networks. To address the research gaps, there is a need to collect real CAVs data, simulate attacks in a controlled situation, and process the network communication data to be suitable to CAV communication.

2.4.3 Anomaly Detection

As CAVs can collect huge amounts of data per second, it is impossible to detect attacks manually. In addition, the driving environment is dynamic, so the detection time requirement is crucial for CAVs. The delay of detection or missing detection can both result in terrible consequences. In satisfying these requirements, IDS (Intrusion Detection System) is an appropriate way to detect attacks during driving. As an efficient way to detect attacks and misuses, IDS has been widely adopted in several fields, including traditional networks [96], wireless sensor networks [97] and the Internet of Things [98, 99]. The concept of IDS has been introduced in computer and network security [100]. IDS is an application or device used to monitor and protect an entire system and its communications. It can help to detect attacks and help manage the whole system. According to different data sources from different parts of the system, the IDS system can be classified as Host-based, Network-based, Hybrid, and Network Behaviour Analysis IDS [101]. In these categories, IDS monitors and detects the behaviours in the system logs or the networks.

There is another method to classify IDS based on different detection methods taken during the process. In this classification, IDS can be classified as signature-based, anomaly-based, or specification-based [102].

Signature-based IDS is also called misuse IDS [103]. It is named due to the fact that IDS is based on previously known attacks and flaws. Signature-based IDS has high accuracy on known attacks without generating large volumes of data [104]. However, the method cannot detect unknown attacks because the new attack pattern does not exist in the known data set. In addition, in a signature-based IDS, the updates of the attack data set can also be a challenge as the attack and intrusion can evolve quickly, which could cost even more time and human sources. Besides, the CAVs contain various ECUs, which make the data format complicated. This complex format shows that it is not efficient to update and maintain a large attack data set when using signature-based

IDS. A typical example of a signature-based IDS system is rule-based IDS. Rule-based IDS needs users to acquire the rules of the data first, which could be completed by designing algorithms or by decisions by experts. Because of this, the limitation of rule-based IDS is obvious. The limitation of experts' knowledge could lead to unsatisfactory results. In addition, the data set needs to be updated frequently, and it is impossible to collect all the possible attack types initially.

The other IDS is anomaly-based. This IDS can monitor the whole system, and analyse the network and activities of the system. It can analyse the normal pattern of the system so that when an attack happens, the IDS can recognise the difference between the attack and normal use. The anomaly-based IDS can help to detect unknown attacks [105]. However, the detection may not be as accurate as signature-based IDS because the anomaly-based IDS is more likely to recognise a normal activity as an attack, which will increase the false alarm rate. In addition, setting up the baseline of normal use is difficult. A large amount of data must be collected to help the system to learn the normal pattern of CAVs. Specification-based IDS is similar to anomaly-based. However, instead of using methods such as machine learning, the specification-based IDSs use specification defined by experts. It could reduce the false alarm rate. However, the time of defining the specifications will be long [106].

Considering the characteristics and requirements of CAVs, the technology is still evolving, and the attackers can always find new ways to attack. Anomaly-based IDS is more suitable for CAV cyber security attack detection than signature-based IDS.

In order to conduct anomaly detection, there are several methods to choose from, including knowledge-based anomaly detection, statistical-based anomaly detection and machine learning-based detection. Knowledge-based detection shares similarities with signature-based IDS. For example, both methods depend heavily on experts' opinions [102]. However, the methods of these two IDS are different. The signature-based IDS requires all the detailed patterns

of attacks while knowledge-based IDS does not. The statistical-based anomaly detection uses statistics to analyse the data, which is quick but it requires accurate statistics distributions [107]. In a real world situation, the data could not be both independent and low dimension as required.

Among these methods, machine learning-based anomaly detection has been widely used, as machine learning methods can fit into different applications by learning normal patterns from collected data. It has lower data requirements since data can be labelled or not, which is more similar to real world collected data. However, currently the amount of machine learning-based anomaly detection on CAVs research is limited. The studies on anomaly-based IDS on traditional computers and networks are reviewed, which could guide CAV IDS research as well.

In [108], the authors proposed a new method to conduct anomaly detection on network traffic data by using artificial neural networks. The experiment data types include traffic data, image data, and system data like log files. The authors found that the proposed method achieved a high accuracy of over 98% when detecting anomalous data. In addition, the false positive rate was below 2% at the same time. Compared with signature-based IDS, this newly proposed method improved detection significantly. Though the anomaly detection of the proposed method was not real time, it still showed the possibility of applying machine learning methods on anomaly detection.

In [109], the authors collected data, including sequence length and window length, to build a user profile. When an attack happened, the system compared the anomalous activities with the collected normal data to classify user behaviours. The experiment also collected empirical data to classify the behaviour. The authors found that the machine learning approach could help to conduct anomaly detection. However, there are still several drawbacks. For example, more algorithms need to be used to achieve higher accuracy. In addition, setting the baseline of normal activities is also a challenge, especially that the normal pattern might change. In order to make machine learning-based

anomaly detection suitable, data needs to be collected on normal behaviours, and the machine learning algorithms need to be suitable for the applications. In [110], the authors emphasised the importance of anomaly detection, which is a fundamental step in securing the information in network communication. In this study, deep neural networks were used to build the model to detect attacks in the NSL-KDD data set, and the trained model was then adapted to the NSL-KDD testing data set to be evaluated. This research aimed to examine the suitability of the deep neural network model because the anomaly detection environment is always changing. The authors found that deep neural networks could help anomaly detection obtain good results. Deep conventional neural networks and Recurrent Neural networks achieved 85% and 89% accuracy respectively on the NSL-KDD testing data set. The authors also believedd that feature selection methods could improve the performance of the models further.

In [111], the authors proposed that the signature-based IDS could not detect unknown attacks such as zero-day cyber attacks. The anomaly detection IDS was then developed to detect this kind of attack. In this paper, SVM (Support Vector Machine) was used as an enhanced unsupervised machine learning approach with a low FP (false positive) rate. In the study, normal packets were collected by Self-organized Feature Map. Because the data is not labelled, unsupervised learning were used. Feature selection methods using the Genetic Algorithm were also used to identify the most relevant features in the collected data. Time frequencies of data were considered during the detection. The study found that the enhanced support vector machine method is effective in classifying attacks in network traffic.

Among all the literature and studies mentioned above, machine learning-based anomaly detection is a useful and powerful way to detect anomalies in the network. Because CAVs are like mobile personal computers in the future, though there are several different characteristics and requirements, the characteristics of traditional networks are still worth learning. In addition, the researchers

also did various work on machine learning-based anomaly detection to detect anomalies on CAVs.

In [112], the authors comprehensively reviewed the current adversarial attacks on CAVs using machine learning algorithms. The potential attacks were divided into several categories, including the application layer, network layer, system level, privacy breaches, and sensors attack, etc. The authors also emphasised that the intrusion detection of cyber attacks is of high importance in the development of CAVs.

In [113], the authors built a scheme based on the machine learning algorithm CatBoost and a Morsel supple filter to predict the location, and detect the jamming attack. With the anti-jamming scheme, the performance of vehicular communication was increased, with better accuracy and a lower packet loss ratio. The authors concluded that the machine learning-based scheme works effectively against the jamming attacks on the CAV location.

The authors in [77] did a comprehensive survey of 65 related papers on anomaly detection research on connected vehicles. They found that most of the studies were performed based on simulated data. The in-vehicle and communication data were considered separately in almost all the reviewed researches. In addition, the baseline of anomaly detection was not clear, which made the evaluation difficult to conduct. Several existing problems with CAV cyber security studies were listed in the paper. Same as mentioned in Section 2.4.2, though most of the researches used simulated data sets, the data sets were still not available online, which also poses a new challenge.

In [114], the authors proposed a new method for detecting the anomalies of driving manoeuvres based on smart phone data and GPS data. The proposed method built a model combining the vehicle, the driver and sensors on mobile phones. The experiment data set was collected from several vehicle models on different driving traces from over 4800 users. The user behaviours were compared with normal driving behaviours in order to classify them. The metrics of dangerous manoeuvres were also introduced in this study. It was found that

this method could help to detect risky driving manoeuvres, and it could be adapted to different vehicles in different driving situations. However, the data was acquired from traditional vehicles. In addition, risky driving behaviours play only a small part in CAVs as CAVs do not require "the driver". Compared with detecting drivers' anomalous manoeuvres, detecting the CAV itself is even more important.

In [115], the authors found that IoVs (Internet of Vehicles) could generate large amounts of driving data, which makes anomaly detection possible. This detection can help to increase vehicle safety. In this large amounts of data, the authors proposed a new online unsupervised approach called SafeDrive to detect abnormal behaviours by setting normal behaviours as a baseline. The behaviours include, but are not limited to, speed, gear, acceleration, and other basic driving functions. To evaluate the performance of the proposed SafeDrive, it was adapted to a system with more than 29,000 vehicles. The results have shown that SafeDrive is an efficient method to detect abnormal behaviours in large amounts of vehicle data.

In [62], the authors proposed a machine learning-based method to protect the connected vehicle from the vehicle network, CAN, and OS. With the extracted data from these ports on the vehicles, the model learned the normal behaviours, which were then used as a baseline to classify the anomalies. The data was collected from a well-known simulator SUMO (Simulation of Urban Mobility). A Hidden Markov Model was trained on the extracted data to learn the normal behaviours. A regression model was then built based on time frequencies to detect the attacks. The authors also defined several terminology, including "event" and "story" to classify the driving scenarios clearly. 4,000 drivers around the city were simulated, and the system recorded the activities of each driver. To enhance the reliability of the simulated data and to simulate an environment like the real world, noise data was also added to the data set. Several types of attacks, including out of order, update attack, communication attack, and malicious updates, were also defined. In the end, it was found that

the built model is effective in detecting attacks, which was also adaptive to other new interfaces.

As described above, currently, researchers have realised the importance of securing CAVs. The researchers also agreed that machine learning-based anomaly detection can be a good and effective solution to cyber security issues on CAVs. However, there still exist several challenges and limitations. First of all, though anomaly detection developed maturely on traditional networks, the research on CAVs is still in its initial phases, and many studies are still carried out on simulated data. The data sets are also not open source online, making the evaluation and comparison of the data sets difficult for other researchers. Moreover, the baseline to evaluate the anomaly detection results is not clear. All these research gaps still need further investigations.

2.4.4 Feature Selection

In order to improve the performance of anomaly detection by machine learning, feature selection is a widely-used solution. The detection accuracy can be improved by using feature selection methods, and the detection time can normally be shortened. In the CAVs application, the high dynamic environments demand fast detection time. Feature selection is irreplaceable in the CAV cyber security framework.

In [116], the authors tried to improve the intrusion detection system by using feature selection methods. The NSL-KDD data set was used in this experiment. Because NSL-KDD has 41 attributes, the feature selection helped reduce computation time. The training set was used to select feature subsets, and then the subsets were evaluated on the testing set. Two feature selection subsets were chosen, selecting 25 and 35 features, respectively. All the feature selection methods achieved lower accuracy and false negative rates as well. The best performance was achieved by a Decision Tree-based classifier with 35 features. The NSL-KDD data set includes various types of attacks. However,

only binary classification was used. Localising the attack precisely in CAVs is highly important, mitigation methods need to be conducted based on it. Therefore, the binary classification is not appropriate for CAV cyber security anomaly detection.

In [117], the authors compared and analysed three different feature selection methods to evaluate the performance of malicious detection. The experiments were conducted on a benchmark data set, KDD99. The authors emphasised that as the data amount is always large in the network, data mining is of high importance. Feature selection methods, including fuzzy rough subset evaluation with the Hill Climber search method, correlation attribute evaluation with the ranker search method, and CFS (Correlation-based Feature Selection) subset evaluation with best first search method, were used on KDD99. After comparing these results to the results without feature selection methods, the study found that the feature selection methods had no significant effects on detection. However, it should be noted that, in this research, the performance was only evaluated on accuracy. The results still need to be further analysed. More experiments and more feature selection methods need to be conducted and used.

In [118], the authors proposed a new hybrid intrusion detection model. The model achieved a high accuracy in a short time. The experiments were conducted on the NSL-KDD data set, and the accuracy of the new model achieved 99.81% and 98.56%, respectively, for binary and multi-class classification. Though the accuracy was high, the false positive rate and false negative rate were not satisfactory, indicating that the model did not perform well. In order to improve the performance of the model on false positive and false negative rate, wrapper feature selection methods were used on Decision Tree, SVM, Naive Bayes and the proposed hybrid model. The study found that the feature selection could improve the performance of the models. In this research, the effects of feature selection methods have been proved from different dimensions, including accuracy, false positive rate and false negative rate.

The authors in [119] had also emphasised the importance of feature selection in improving the performance of models. As there is an increasing demand for the network, security issues are increasing as well. In this study, the authors proposed an intrusion detection system with adapting feature selection methods. The KDD99 data set was also used in this research. As there are 41 attributes in the data set, not all the attributes are useful for attack classification. In order to increase the efficiency of the model, which is the Decision Tree in the research, the most relevant attributes need to be selected. A new hybrid feature selection method combining the linear correlation coefficient and the cuttlefish algorithm named FGLCC-CFA was proposed, which selected the 10 most relevant attributes from the data set. Compared with the original linear correlation coefficient algorithm and the other three feature selection methods, the proposed feature selection method increased accuracy and decreased the false positive rate at the same time.

In [120], the authors pointed out that there are many irrelevant and redundant features in the data set, which increase the computational burden during the intrusion detection process and the time necessary to carry it out. Recent studies have shown that a comprehensive intrusion detection system should contain a reliable machine learning model and a powerful feature selection method. In the paper, the authors introduced a feature selection algorithm called Flexible Mutual Information Feature Selection, which reduced redundant features. The new feature selection method was then used with a least square SVM method to conduct the intrusion detection on data sets including KDD99, NSL-KDD and Kyoto 2006+. This study found that the combination of the proposed feature selection method and machine learning model achieved better performance on criteria, including accuracy, false positive rate and F-measure, compared to other intrusion detection models.

Among all the researches mentioned above, it could be found that feature selection methods have positive effects on the performance of machine learning models. It speeds up the model to achieve faster runtime with little impact

on accuracy. In addition, it can reduce the false positive rate. These qualities indicate that feature selection is irreplaceable in intrusion detection.

However, in all the studies, there still exist several research gaps. First of all, although there are various papers discussing the use of feature selection methods to improve the performance of machine learning models, the real use of feature selection methods in a real CAV cyber security environment is still missing. Compared with existing studies, the results could be different when adapting to CAVs.

In addition, as there exists a large number of feature selection methods, some researches only discussed a few types of the methods. In CAVs anomaly detection, more feature selection methods need to be assessed. The comparison of more feature selection methods could help to build a more robust intrusion detection system for CAVs. Moreover, CAVs intrusion detection requires even higher performance than normal network traffic data, indicating that the processing time of feature selection methods also needs to be considered. All these research gaps should be considered during the experiments that form the research for the thesis.

2.5 Summary

In this chapter, related papers and researches on CAV cyber security have been carefully reviewed. First of all, the concepts and the current CAVs researches have been introduced. Then, the attacks to CAVs were also discussed, including commercial companies such as JEEP and specific vehicle hackers such as Tencent Keen Lab. To prevent CAVs from being attacked, researchers have tried to define possible attacks and take corresponding reactions to them.

To build a comprehensive CAV cyber security framework, process of risk assessment, related data sets, anomaly detection and feature selection, were reviewed, respectively. It was found that risk assessment is the initial step to build a cyber security framework, as it defines the potential attacks and as-

sesses the severity. Besides, there exists no open CAV cyber security data set, which is a major research gap now. Relevant researches on the intrusion detection system and anomaly detection were then reviewed. Studies also showed that the machine learning-based anomaly detection method is an efficient and automatic way to detect the abnormal behaviour of vehicles, which can also help to detect unseen attacks on CAVs. In addition, the feature selection methods were reviewed because they can improve the performance of anomaly detection.

It could be seen that there is not enough research related to CAV cyber security. After reviewing relevant attacks and papers, several research gaps have been identified as follows to build a comprehensive CAV cyber security framework, which could be summarised as "3D", including "Definition", "Data", and "Detection".

1. Definition: Though some studies listed potential attacks to CAVs, most of these researches were theoretical. In most of the literature, the concepts of CAV, connected vehicles, smart vehicles, and IoT vehicles were used interchangeably. The difference between CAVs and other concepts needs to be clarified to support further research on CAV cyber security. The definitions and terminology of cyber security of CAVs were not well-defined, and it is urgent to define the concepts. Meanwhile, in addition to the concepts, the severity of different attacks needs to be clarified. In most of the research, there are only descriptions of possible attacks, but no systematic ways to evaluate the possibility or consequences of the attacks. Defining the severity of each attack could help researchers prioritize the possible attacks, and make corresponding reactions better.

2. Data: Data is of high importance in CAV cyber security research, since without data, anomaly detection could not be trained or conducted. After surveying the studies, it could be found that open source CAV cyber security data is difficult to acquire, due to confidential reasons. It is also difficult to conduct an attack in the real world environment and collect data, due to

safety reasons. Thus, most of the studies used simulated data. Meanwhile, the contents of the data are also worth investigating. Collecting specific CAV cyber security data is a major research gap in current research.

3. Detection: After reviewing the studies, it could be found that there were several attempts applying the machine learning method to detect attacks. However, the evaluation metrics and baselines of the machine learning models need to be clarified. In addition, because the data is multi-sourced on CAVs, different machine learning algorithms may be suitable for different application scenarios, which also needs to be investigated. Meanwhile, feature selection methods have been identified to be effective in improving the performance of detection models. The effectiveness of feature selection to different models also need to be evaluated in real usage.

In the following chapters, the research gaps mentioned in this chapter will be addressed. A systematic CAV cyber security framework will be built addressing the "3D" research gaps. Definitions of CAV cyber security will be introduced, severity of each attack will be assessed, and relevant data will be collected within the framework. In addition, machine learning-based detection mechanisms will be proposed and evaluated in the new CAV cyber security data sets. The performance will then be improved by feature selection methods. The best machine learning models and important attributes will also be suggested through the experiments.

Chapter 3

CAV Cyber Security Framework

3.1 Overview

To establish a solid foundation of the CAV cyber security framework in the thesis, terminology of CAV cyber security is defined, followed by the investigation of potential attacks. This chapter begins by analysing and gaining an understanding of UK CAV cyber security principles [15]. In the thesis, the total eight principles are categorised into three stages: before, during and after the attacks. The definitions of CAV cyber security and other relevant terminology are then proposed to provide insight into what CAV cyber security is and what CAV cyber security contains. This could help to build the foundation of the new framework and also address the identified research issues. Based on the definitions of CAV cyber security terminology, a UML based CAV structure is built to help identify the vulnerabilities and potential attacks to CAVs. A more detailed list of potential attacks is then presented, based on which CAV attack severity criteria are proposed to rank the potential attacks in CAVs. To mitigate the attacks, five mitigation methods are suggested, which would be further explored in the following chapters by CAV cyber attack detection.

The main content of this chapter has been published in the Journal of Advanced Transportation, titled "Towards a Severity Assessment Method for Potential Cyber Attacks to Connected and Autonomous Vehicles", in September

2020.

3.2 UK CAV Cyber Security Principles

In June 2017, the UK government published an official document, "Key Principles of Vehicle Cyber Security for Connected and Automated Vehicles" [15]. In this document, the UK government published eight principles in CAV cyber security, covering the whole life cycle of CAVs, and providing guidance of protection from sub-contractors, suppliers and potential third parties on hardware, software, and data. This document emphasises the importance of research on CAV cyber security and highlights the necessity of this thesis. These eight principles are summarised and categorised in this thesis from [15], the structure of which is presented in Figure 3.1.

As it can be seen in Figure 3.1 that Principle 1 is the most important and fundamental, as it defines the requirements of top level design concerning CAV cyber security. In addition to Principles 1.2 and 1.3 that consider human factors, Principle 1.4 considers the design during the research stage. Among all four sub principles, Principle 1.1 on security program is the focus of the thesis, which divides the protection process into three stages:

1. Before the attacks happen. Relevant organizations and manufactures need to define what kind of attacks could happen and their mitigation methods.

2. When the attacks happen. The system could monitor the whole CAV, and detect attacks as soon as possible. The system should also be robust enough to face attacks.

3. After attacks happen. The system could respond to attacks appropriately and recover from attacks.

In the current literature, there is no widely adapted framework in CAV cyber security [121], based on which attack points could be defined and efficient protection methods could be developed. According to the UK CAV cyber security principles categorized in Figure 3.1, the most fundamental parts of CAV cyber

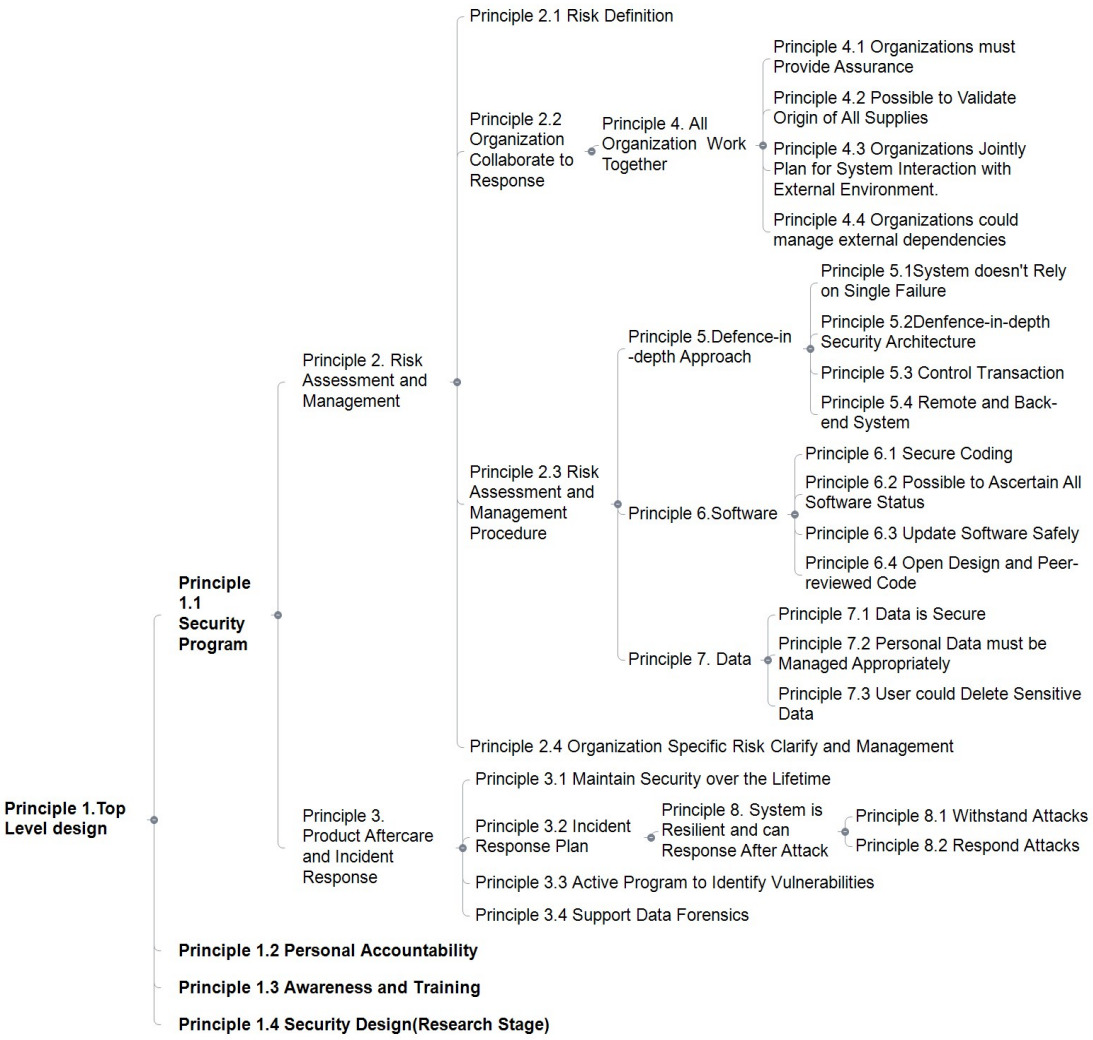


Figure 3.1: *UK CAV Cyber Security Principles Structure*

security are defence-in-depth approach covering physical, technical and administrative controls (Principle 5), software (Principle 6), and data (Principle 7). Before the cyber security attacks happen, risks of the CAV system could be defined, assessed and managed (Principles 2.1 and 2.3); During the operation of CAVs, monitoring the CAV system could help to maintain security over the lifecycle (Principles 3.1 and 3.3). The CAV system could also respond and support effective solutions appropriately after an attack (Principles 3.2 and 8).

The CAV cyber security could thus be divided into the security of hardware, software and data. Besides hardware, software and data generated by CAVs, CAVs are also connected to the outside world via data exchanges with

other vehicles, infrastructures, or pedestrians, which makes the communication channel an attack target as well. The relationships between these components also need to be defined.

3.3 CAV Cyber Security Terminology and Definitions

With the CAV cyber security principles, various terminology and definitions need to be provided, which are quite often vague in the existing literature. Because of the unique characteristics of CAVs and the dynamic driving environments, cyber security definitions in computer science can not be used for CAVs directly. Although the UK government has published CAV cyber security principles to strategically guide the development of CAV cyber security, the terminology still needs to be defined more clearly. Even in the field of computer science, security-related terminology, such as information security and cyber security, are misused frequently [122]. To conduct interdisciplinary research across the fields of the CAVs and cyber security, it is essential to offer a universal definition of cyber security for CAVs. Relevant terminology and definitions within the CAV cyber security framework are provided in this section.

Parts of the content in this section has been published on the Forum on Cooperative Positioning and Service (CPGPS), titled "Survey on cyber security of CAV", in May 2017. This paper has also been selected as the best student paper award on the Forum.

CAV Cyber Security (CAVCS): CAV cyber security protects the CAV Network from attacks, intrusion, interruption, damage or unauthorized usage by taking appropriate protection methods. The aim of CAVCS is to maintain the stability of the CAV Network and maintain the confidentiality, integrity and availability of the data in CAV Network. The personal safety of users thus

could also be guaranteed. This terminology is modified from the well-adapted computer cyber security definition [123] and US National policy [124].

CAV Network: CAV Network is formed by nodes including CAVs and other information terminals such as smart infrastructures, the cloud platform and pedestrians, etc. These nodes collect, store, transmit, exchange and process data to retrieve information. The edges in the CAV network represent the communication between these nodes. These different statuses of nodes and communication form the dynamic driving scenarios.

CAV Data: CAV data contains all the data collected, stored, transmitted, exchanged and processed in the CAV Network. Terminals could retrieve relevant information from the processed data to help to make driving decisions. This terminology is based on [125], which is a detailed description document of UK CAV cyber security principles. The more detailed data types are listed in Section 3.6.

In addition, vulnerabilities, threats and attacks are used interchangeably in several descriptions. However, the concepts are different distinctly [122]. Defining and understanding the differences between the three concepts could help to a clear and precise understanding of CAVCS. The relationships between vulnerability, threat and attack are shown in Fig 3.2.

CAVCS Vulnerability: Vulnerabilities are the potential flaw or weakness in CAV design and implementation. Vulnerabilities, if exploited, will lead to threats to the CAV system and network.

CAVCS Threats: Threats are potential attacks to the CAV system and CAV network. Vulnerabilities will expose the CAV system to potential threats to the system and network.

Figure 3.2: *Relationship between Vulnerability, Threat and Attack*

CAV Cyber Attack: CAV cyber attacks refer to actions to attack, intrude, interrupt, damage to the CAV system. It could also include unauthorised access to the CAV network to monitor or eavesdrop, and steal information from data. The definition of CAV cyber attack is derived from the US National policy [124]. The CAV cyber attacks could be categorised into two main types, namely passive attacks and active attacks [63]. More detailed potential attacks are investigated in Section 3.6.

Passive attacks include eavesdropping or monitoring the transmissions between users, where the attackers cannot modify or change the content in the transmission, and would not interact with the data transmitted [126]. Passive attacks most probably faced by CAV include eavesdropping, the release of the information and traffic analysis. Normally, passive attacks are difficult to identify because the attackers do not modify the contents in the communication data. Therefore, to deal with passive attacks in CAV, defending and protecting is more important than detecting passive attacks.

Active attacks are to modify or jeopardize the messages and the data transmitted [127]. These could cause much more severe damage than passive attacks, especially in the CAV environment, and even cause fatal injuries. In CAV, active attacks include spoofing, reply attacks, modifications and DoS (Denial of Services). Detection is effective to mitigate this kind of attack.

3.4 UML-based CAV Structure

Unified Modeling Language (UML) is widely used in software engineering to define and model the structure of a system [128]. In UML, the class diagram is used to build the concept structure of a system, showing both main components and their relation with other components in the system.

In Figure 3.3, the UML based CAV cyber security framework is developed to define the relationship between each component and the structure in CAV, including hardware, software and their generated data, which help vehicles

function well. Based on the framework, different types and points of potential CAV cyber attacks can be analysed and categorised. The main classes in this UML based CAV framework include Vehicle Data, Data Processor and Vehicle Functions.

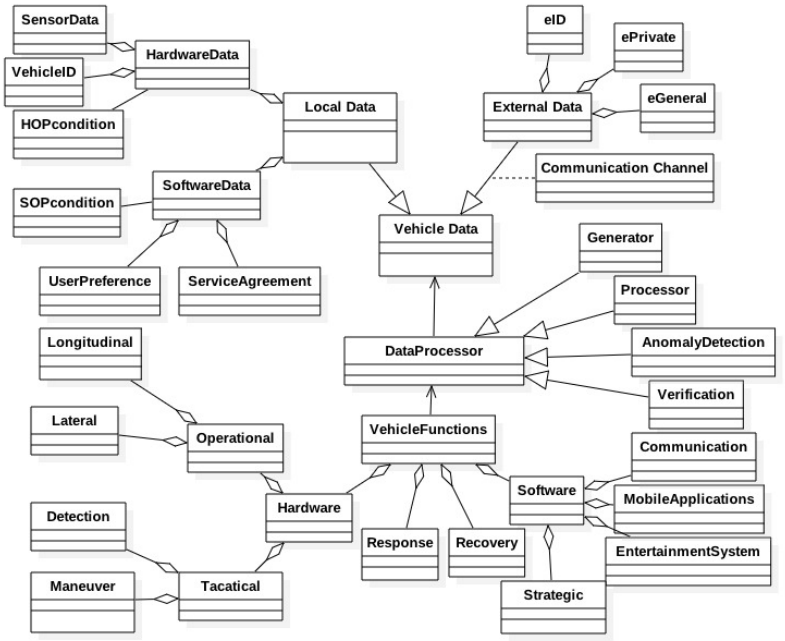


Figure 3.3: *UML based CAV Structure*

3.4.1 Vehicle Data

CAVs make decisions and implement relevant vehicle functions based on data, thus Vehicle Data presents the most fundamental component in the CAV structure. In the Vehicle Data class, the data could be divided into local data and external data. Class Vehicle Data refers to Principles 5 to 7 in Figure 3.1.

Local data has two sub-classes, which are hardware data and software data. These two sub-classes include not only data generated by hardware and software, but also the operation condition data of hardware and software. Class HardwareData is the sensors data collected from vehicle surroundings by various CAV sensors including Radar, GNSS(Global Navigation Satellite Systems), and camera [129], for example, GNSS and image data which determine the current position of CAVs. In addition, VehicleID contains the data to identify the

vehicle, such as the electric plate (a unique number or letters assigned by the government department). As CAVs exchange data and information with other entities, including other CAVs, infrastructures and pedestrians, VehicleID also contains a unique pair of a public key and private key, which will be used to encrypt messages and check the identification of vehicles [130]. Class HOP-condition is the operation condition data of hardware.

Class SoftwareData in Local Data includes the data collected by the software in CAVs, such as on-board entertainment system. CAVs will be very likely an important smart mobile device people use in the future [131]. They not only provide decision support or solutions such as the shortest driving route from place A to place B, but also service users' preferences such as 'the most beautiful route', or 'the quietest route'. Class UserPreference contains such preference data of users preferences, based on which CAVs make the best decision for the specific users. Class ServiceAgreement defines protocols that the software will comply, including privacy protection and other services protocols. Class SOPcondition is the operation condition data of software.

Class External Data is received from other entities such as other CAVs and intelligence infrastructures in the CAV network. All the data is received via communication channels such as V2V (Vehicle to Vehicle), V2I (Vehicle to Infrastructure) communication, which is the Class Communication Channel. As each entity has its own ID stored in its local data, the external data will also need this information to guarantee the legal identification of the data sender, and Class eID contains senders' ID information. In external data, after identifying senders' ID, messages will be divided into either private or general, based on Principle 7.2 which states that data should be managed appropriately. In certain scenarios, vehicles or infrastructures need to send private data such as users' preferences. This could only be accessed by specific users and is stored in Class ePrivate. Class eGeneral stores data that everyone could access, such as position data and vehicle size data.

3.4.2 Data Processor

CAVs deal with a massive amount of data every day. It is reported that each CAV on itself produces up to 4000 GB from just one-hour driving per day [132]. In addition, adding a V2V communication system to a vehicle may require 10 messages per second [133], which will also increase the data processors workload. How the data is processed is even more important than how it is collected. CAVs are equipped with a data processor to clean data and support making appropriate decisions. Class Data Processor is related to Principles 2.3, 3.1 and 3.3 in Figure 3.1.

Class DataProcessor contains four basic data processing methods, and this class relies on the Class Vehicle Data. Class Generator gathers data from different sources, and the formats from multiple data sources need to be regulated and fused for processing. Class Processor processes the data, including cleaning or annotating the data for analysis. Class Verification includes components that ensure the data is secure, fulfilling the cyber security requirements in the CAV system. During these processing steps, the CAV system also needs to be able to detect abnormal situations in hardware, software and data. Class AnomalyDetection detects any such vulnerabilities and anomalies in the CAV system.

3.4.3 Vehicle Functions

If there is no anomaly behaviour in the CAV system, relevant data will be used to make decisions using Class Vehicle Functions after being processed. Class Vehicle Functions is related to Principles 3.1, 3.2, 5, 6 and 8 in Figure 3.1, and is defined accordingly as shown in Figure 3.3.

The functions of CAVs could be divided into Classes Hardware and Software in the CAV structure, as shown in Figure 3.3. A variety of different driving tasks and operations have been categorized based on SAE J3016, which was introduced in Section 2.2.

After a CAV detects its surrounding objects, it uses operational functions to respond. Based on SAE J3016, Class Hardware is divided into Class Operational and Class Tactical. Class Operational has two sub-classes, which are longitudinal and lateral. These two sub-classes include relevant hardware functions when the vehicles are in longitudinal or lateral motions. Class Tactical also has two sub-classes. Class Detection is to monitor objects and events around through sensors including Radar, Lidar and cameras. Class Manoeuver is to take relevant manoeuvers such as turning the indicators on.

Beside hardware functions, software functions, such as entertainment system and mobile applications, are also essential parts of CAVs. In addition, Class Communication supports all the receiving and sending data functions. Class Strategic is to plan the whole trip including the best route, travel time and destinations, which is defined based on the strategic functions in SAE J3016.

In addition to Classes Hardware and Software, Class Response takes relevant actions based on the data from the hardware and software. Class Recovery is to fallback when a system failure happens, making sure CAVs are resilient and fail-safe.

3.4.4 Possible Attack Points

Cyber attacks in computer networks could be categorized into different types including viruses, worms, buffer overflows, DoS attacks, network attacks, physical attacks, password attacks and information gathering attacks [134]. In traditional automobile vehicles, the points of attacks have been categorized into two types [135], namely attacks to the audio system or mobile applications, and attacks to CAN (Controller Area Network), which is an inner vehicle communication network for micro-controllers and devices. As CAN is connected to all the in-vehicle hardware components including brakes, air conditioner, steel and wheels, the second type of attacks is more dangerous than the first one.

Compared with computer networks and traditional automobiles, CAVs are equipped with both physical parts and software, and they are connected within the overall transportation infrastructure, thus all the above attacks to automobile could happen to CAV. Moreover, with the increasing number of autonomy and connectivity functions, there will be more vulnerabilities or attack points on CAVs. As it is described in Section 3.3, CAVCS is to protect the whole CAV network from cyber attacks affecting the performance remotely or physically to guarantee the personal safety. It is necessary to identify, define and classify possible types of attacks to CAVs at early stage. Based on the UML based CAV structure established in Figure 3.3, four types of possible CAV attacks and sub-attacks are listed below.

1. Vehicle physical parts. These CAV physical parts include the windscreen, wheels or even brake. It is already reported that hackers could take control of brakes or air conditioners on Nissan [136] and JEEP. JEEP even recalled more than 1.4 million vehicles to install security patches due to this type of hacking [137]. The attacks towards hardware may be conducted physically or remotely. The attack methods including misleading the hardware to make wrong driving decisions, or hacking into the hardware to eavesdrop the activities.

There are several attack points on the CAVs hardware. The mainstream sensors on CAVs include cameras, Lidars (Light Detection and Ranging) and radars as included in Table 3.1. All of these sensors could be attacked physically or remotely. For example, the cameras would be misled by fake images or the radar signal could be jammed. Attackers could even hack into the camera system to monitor the vehicle's activities. Moreover, the GNSS system would also be attacked by experienced attackers. For example, the GNSS system could be jammed, and then the vehicle may not receive the GNSS signal to navigate and locate its position.

2. Vehicle software. CAVs could be installed with more than 100 million lines of code, while a Boeing's new 787 dreamliners is equipped with only 6.5 million lines of code [138]. This leads to a larger number of vulnerabilities in

CAVs. The entertainment system, the installed mobile applications, and the audio system onboard, could all be potential attack points for attackers. After taking control of the software, the data exchange could be monitored, or even the hardware could be harmed if software is taken control.

3. Data. CAVs data stored on the vehicle is transferred between CAVs, vehicle to infrastructure or to pedestrians and cyclists. Attacks to data, including local vehicle data such as vehicle ID including electronic platform or vehicle model, personal data like users preferences, could lead to data leakage. In addition, because CAVs may support payment services, such as toll service, private data such as payment transfers could also be an attack point in CAVs. External data received from other users in the communication range would also be attack points. Modification on communication data or injecting fake messages will cause not only problems of information leakage but also traffic congestion or even collisions.

4. Communication channel. The potential attacks may also target the communication channels. The attack points can be via V2V (Vehicle to Vehicle), V2I (Vehicle to Infrastructure), V2C (Vehicle to Cloud) and V2X (Vehicle to Everything) communication. The communication channel would be easily blocked if attackers send huge amount of messages at the same time. In addition, eavesdropping communication channels would also cause information leakages.

Based on the analysis, the possible attack points to CAVs are summarized in Table 3.1. As the technologies adapted on CAVs are still evolving, these attack points will definitely increase in the future. However, as the attack points are within the scope of physical parts, software, data and communication channels, the table is extendable to include and categorise different types of new attacks

In this thesis, more detailed attack points will be analysed and assessed in Section 3.6.

Table 3.1: Possible Attack Points to CAVs

Category	Attack points
Physical Parts	Sensors(LiDAR, Radar, Camera), GNSS device, vehicle system (OBD, CAN-bus, power system) etc.
Software	Mobile applications installed on the vehicle, in-vehicle system (entertainment system), data processing system, decision making system etc.
Data	local data (vehicle ID, payment information, userś personal information), Exchange data (Vehicle's speed, brake status) etc.
Communication Channel	V2I (Vehicle to Infrastructure), V2V (Vehicle to vehicle), V2C (Vehicle to Cloud), V2X (Vehicle to everything) etc.

3.5 A New Severity Assessment Method of CAV Cyber Security

The potential attack points or attack ports are analysed firstly. For each potential attack, the following criteria will then be evaluated to define the severity of the attack.

The criteria are chosen based on a well-adapted formula in engineering risk assessment for transportation and infrastructure [139], information technology system [140], and civil aviation [141], as is shown in Equation 3.1:

$$Risk = Asset * Vulnerability * Threat \tag{3.1}$$

According to the formula, the criteria are divided into three parts, namely asset for the possible attack targets, vulnerability for the possible risks to the attack targets, and threat for the possible consequences. However, as it is mentioned in Chapter 1, there are several differences between traditional automobile network cyber security and CAV cyber security, only appropriate criteria are chosen here. For example, to evaluate the severity of the risk, CAVs should consider not only information leakage level but also physical damage level.

Asset:

1) Asset Name: In computer security, ISO/IEC 13335-1:2004 defines that assets contain all the hardware or software parts on computers that could be an attack target such as a data set, one piece of hardware or software code [122]. In CAVs, as there are lots of ECUs and sensors on the vehicle, there will also be an abundance of possible targets. More detailed assets will be introduced in Section 3.6.

2) Asset Importance: The importance of each asset is divided into three levels:

a. Low: The breakdown of this asset will not affect the operational function and tactical functions of the whole system. In the SAE J3016 standard [36], operational functions includes lateral and longitudinal vehicle motion control. The operational functions are the most basic functions of a vehicle, which include starting, stopping, driving and controlling [142]. The tactical functions contain the OEDR, which are introduced in Chapter 2.

b. Medium: The breakdown of this asset might influence tactical functions of the vehicle. But the breakdown would not have direct effects on the operational functions. In addition, the asset function could be replaced by other assets on the vehicle. For example, if cameras on CAVs break down, the vehicle could still use other sensors to detect the surroundings.

c. High: The breakdown of this asset may cause damage to operational functions of the vehicle directly. For example, the in-vehicle system, which could send instructions to ECUs to maintain the vehicle speed or stop the vehicle in needed situations, is of high importance.

Vulnerability:

1) Risk Name: To each asset, there may be more than one risk. In this criterion, specific risks to each asset will be analysed, more details are presented in Section 3.6.

2) Difficulty of Conduction: The difficulty of conducting an attack varies regarding the attack characteristics. Some attacks may require attackers with sufficient knowledge in specific areas such as GPS spoofing or fake identifi-

cation. Some devices are securely protected such as GNSS satellites, which are protected by the governments. Hacking into these devices may need not only knowledge but also sufficient time and money. The Difficulty of conduction is considered from knowledge, time and budget needed. The difficulty of conduction are then graded into three levels listed as below.

a. Low: Attackers do not need to acquire relevant knowledge to conduct the attack or the target asset is easy to be obtained/bought on the market. The attack is not time-consuming.

b. Medium: Attackers only need to spend a short time (weeks/months) to learn the required knowledge. Hacking into the target asset needs to be purchased at a high price, or the hacking process is time-consuming.

c. High: Attackers need to have extensive knowledge on the target asset or need to spend years to learn relevant knowledge. The target asset is difficult to find in the market or costs an astronomical figure.

3) Detection Possibilities: This criterion is to define the level of detection possibilities of attacks by the users or the system on the vehicle. In computer science, the attacks are divided into two main categories, namely passive attacks and active attacks [143]. Passive attacks will not interrupt the system but will monitor or eavesdrop it to steal information. Active attacks will interrupt the system functions directly by methods such as injecting fake messages. In general, passive attack is difficult to detect but easy to defend, while active attack is difficult to defend but easy to detect [63]. Though passive attack may not cause harmful effects on system functions, the information leakage could also be a severe risk because CAVs will be the ultimate personal mobile devices in the future [144], which means they could store sensitive data including personal home address, contact numbers and financial information. Based on this, it is essential to know the detection possibilities of different attacks. The levels of detection possibilities are divided into three levels as listed below.

a. Low: The attacks will not affect any function (whether operational or tactical functions) of the CAV system. It is difficult to detect the attack in

normal use. The best solution is to prevent the attacks from happening in advance with encryption or authentication.

b. Medium: The attacks will not affect the operational functions of CAV system so the users would not notice the attacks immediately. But the attacks would affect some parts of the tactical or strategic functions. The system will detect the abnormal behaviour afterwards and warn users.

c. High: The attacks will influence the operational function immediately so the users could notice the attacks immediately. For example, if the vehicle suddenly stopped on the road, the users would notice the abnormal situation immediately. In addition, if the cameras around the vehicle break down, the system will notice this abnormal situation promptly.

Consequences:

1) Consequence Name: To each possible risk, there may be more than one consequence. The consequences will be listed and then be analysed, more details are presented in Section 3.6.

2) Severity of Information Leakage: Information leakage has been a cyber security problem in the field of computer science. Information leakage attacks usually damage the confidentiality, integrity and availability of the system [122]. The severity is based on the scale and importance of the leaked information.

a. Low: The attack will not leak any private information.

b. Medium: The attack will leak small-scaled personal information or unimportant information not related to confidential information. For example, the attacker would know the preference of the passenger on choosing route or on the entertainment system. This type of information leakage will not cause further harm directly.

c. High: The attack will leak high-importance confidential information such as the financial information, the home address or the personal ID. By knowing this information, the attackers could conduct further harmful actions to the victims. In another situation, this information leakage would cause larger scale

information leakage such as personal data stored in the cloud.

3) Severity of physical damage: Compared with traditional networks, CAVs could lead to physical damage to people directly. Tesla vehicle has already caused fatal incidents on a straight road with good visibility in a good weather [145]. On March 2018, a Uber autonomous driving vehicle struck and killed a pedestrian crossing the road in Arizona, US [146]. The Uber test vehicle failed to detect the pedestrian in the low visibility environment and didn't conduct any corresponding actions. As a big metal machine, the CAV could cause hazards or even be exploited as a weapon. Based on the possible consequences, the severity of physical damage are graded as below.

a. Low: The attacks are not likely to cause physical damage to human or other vehicles, and infrastructures.

b. Medium: The attacks are likely to cause small hazards. These hazards could cause damage to infrastructures or vehicles, but would not cause fatal injuries to people.

c. High: The attacks have a high possibility to cause fatal injuries incidents.

4) Combined Severity Level: The method to evaluate the combined severity is adapted from the risk management in information system [140]. In information system, the risks are determined by the likelihood and impact. To the combined severity levels to CAVs, a new severity matrix is built based on the severity of information leakage and physical damage. It is shown in Table 3.2, which is adapted from [140].

In this table, if the severity of information leakage and physical damage are in same level, then the combined severity will be in same level as well. For example, if the severity of information leakage and physical damage are both low, then the combined severity of this risk will also be low. However, considering the importance of severity of physical damage, if the severity of physical damage is high, the combined severity level will be high as well.

5) Recovery Time: This criterion is to evaluate the time needed to recover to normal situation after the attack has been detected.

Table 3.2: Combined Severity Level Matrix for CAVs

Information Leakage/Physical Damage	Low	Medium	High
Low	Low	Low	High
Medium	Low	Medium	High
High	Medium	High	High

a. Low: After the detection, the damage caused by the attack could be fixed in a timescale of seconds to minutes.

b. Medium: After the detection, the damage caused by the attack could be fixed in a minutes to hours timescale.

c. High: After the detection, the damage caused by the attack could be fixed in a timescale of hours to days.

Based on these criteria, possible attacks in different scenarios are analysed in Section 3.6. It should also be noticed that this thesis aims to discuss the possible cyber security attacks to a full CAV (Level 5). It indicates that all the possible attacks could be conducted through wireless communication remotely. In this case, the physical access of attacks were not considered when evaluating the severity. These criteria may not be comprehensive and exclusive, and some parts could be further considered and investigated. This thesis presented the first attempt to define and rank the possible attacks severity in CAV scenarios. This would also be a starting point to raise public and CAV practitioners' awareness towards CAVCS.

3.6 Possible Attacks and Severity Assessment

In this section, possible attacks of CAVs will be listed. Following the criteria defined in Section 3.5, severity of each attack will be analysed.

Detailed potential attacks will be analysed from these two aspects namely the connectivity and automation covering the in-vehicle and inter-vehicle components including hardware, software, and data in CAVs. The list of potential attacks is presented in Table 3.3.

Table 3.3: Possible Attacks on CAVs

	Asset Name	Importance of this Asset	Risk Name	Conduction Difficulty	Detection Possibilities	Consequences	Information leakage	Physical Damage	Severity Level	Recovery Time
Auto -mation	Audio/Video Devices	Low	Loud Volume	Medium	High	Distract passenger's attention	Low	Medium	Low	Low
			Fake sound	Medium	High	Cause passenger's panic	Low	Low	Low	Low
			Remote control other parts in vehicle	High	Medium	Take control of the vehicle	High	High	High	Medium
	Cameras	Medium	Blind Vision	low	High	block the vision of vehicles	Low	Medium	Low	Low
			Mislead Cameras (fake vision)	low	Medium	wrong reactions of vehicles	Low	Medium-High	Medium	Low
	Battery System	High	DoS	High	Medium	Consuming battery energy	Low	High	High	High
	Lidar	Medium	Jamming	Medium	Medium	decreasing the performance of LiDAR	Low	Medium	Low	Medium
			Hidden Objects	Medium	Medium	wrong reactions of vehicles	Low	Medium - High	Medium	Medium
			Fake Objects	Medium	Medium	wrong reactions of vehicles	Low	Medium	Medium	Medium
	Radar	Medium	Jamming	Medium	Medium	decreasing the performance of Radar	Low	Medium	Low	Medium
			Hidden Objects	Medium	Medium	wrong reactions of vehicles	Low	Medium - High	Medium	Medium
			Fake Objects	Medium	Medium	wrong reactions of vehicles	Low	Medium	Medium	Medium
	GNSS	Medium	Spoofing	High	Low	wrong position	Medium	High	High	High

Table 3.3: Possible Attacks on CAVs

	Asset Name		Importance of this Asset	Risk Name	Conduction Difficulty	Detection Possibilities	Consequences	Information leakage	Physical Damage	Severity Level	Recovery Time
				Jamming	Medium	Low	Signal lose	low	Medium	Low	Medium
	in-vehicle system		High	Injection	Meidum	Medium	Information leakage and wrong reaction	Medium	Medium	Medium	Medium
				Eaves -dropping	Medium	Low	Leakage of personal information	High	Low	Medium	Low
				Traffic analysis	Low	Low	Leakage of daily usage	High	Low	Medium	Low
				Modification	High	Medium	wrong message received	Medium	Medium	Medium	Medium
Connecti -vity	with Other vehicles		Medium	DoS	Low	High	Block vehicle communication channel	Low	Medium	Low	High
				Modification Message	Medium	Medium	wrong reactions of vehicles	Medium	Medium	Medium	Medium
				Fake/ghost message	Medium	Medium	wrong reactions of vehicles	High	Medium	High	Medium
	With In -fra -structure	Infra -structure	Low - Medium	Change infrastructure Sign	Low-Medium	Medium	wrong reactions of vehicles	Low	Medium-High	Medium	Low
				block/remove sign	Low	Meidum	No reaction in specific position	Low	High	Medium	Low
		Road	Low-Medium	Road line /infrastructure Changing	Low	High	wrong reactions of vehicles	Low	Medium	Low	Low
	Cloud system	authority	Medium	Fake Identity	High	Medium	Identity leakage	High	Medium	High	Medium
		Cloud data set	Medium	Injection	High	Medium	wrong message received	Medium	Medium	Medium	Medium-High

Table 3.3: Possible Attacks on CAVs

	Asset Name	Importance of this Asset	Risk Name	Conduction Difficulty	Detection Possibilities	Consequences	Information leakage	Physical Damage	Severity Level	Recovery Time
			Modification	High	Medium	Wrong information from cloud	Medium	Medium	Medium	Medium-High

Automation:

Different autonomous levels may have different attack possibilities. According to the SAE automation level [36], this thesis will focus on the attacks to a fully automation vehicle (level 5). It indicates that the CAV is capable of all the DDT under all circumstances. It is also assumed that all the vehicles on the road are CAVs. In real world situations, there will be a mix of fully CAVs and traditional vehicles for certain period of time. In addition, it is obvious that CAVs will keep evolving and more technologies will be adapted. This thesis only discusses attacks with existing CAV technologies. However, as the attacks are categorized on automation and connectivity in-vehicle and inter-vehicle, the list of possible attacks could be extended if new technologies are adapted to CAVs.

In current CAV development, all the vehicles from different companies have installed multiple sensors. The mainstream sensors include LiDAR, Radar and camera [147, 148]. For example, Google Waymo vehicles are installed with a 360-degree camera on the roof as the vision system, several LiDAR sensors and Radar sensors around the vehicle body [149]. There are also supplemental sensors such as sound detection sensors.

The possible attack target assets are listed below.

1) Audio/Entertainment Devices: Audio devices have already been used in modern automobiles widely. It evolved to a colourful touchable screen showing more information in vehicles [150]. In CAVs, the audio/video system could be used to warn users about anomaly or abnormal behaviours detected in the system or surrounding environments.

a. Loud volume. The first possible attack is to suddenly increase the volume of the voice such as background music on board. This attack could distract the passengers' attention. Sudden loud sound could even cause passengers' panic in certain situations. The severity of information of leakage is low but the severity of physical damage is medium, which means that the overall severity is low.

b. Fake sound. The attacker could use the audio system to make fake noise such as crash sound. This attack might cause passengers' panic as well. But this attack is not likely to cause information leakage.

c. Remote control. This attack already happened in real world environment. Two white hackers in USA hacked into the entertainment system on a Jeep Great Cherokee from 10 miles away and then stopped the vehicle on a highway road through the entertainment system [136]. This is because the vehicle CAN and entertainments system are combined together. If an attacker could control the vehicle remotely through the audio/entertainment system, the severity of physical damage could be high. In addition, the risk of information leakage will also be severe because the attackers could send remote instructions to gather private information. Moreover, in CAVs, the remote control attacks might happen on other components of the vehicle, which makes the attack severe.

2) Cameras: Cameras provide the vision data, an indispensable part in CAVs. To detect the surrounding objects and position the vehicle, camera is a fundamental sensor on CAVs. However, the camera's function could still be replaced by other sensors if cameras break down, thus camera is of medium importance. There are successful attacks to cameras to fool vehicles already [151].

a. Blind vision. Blind vision attack could be easily achieved by physical access. However with the connectivity of the vehicle, it is even easier for the attackers. The attackers could disable the camera by controlling a strong light resource remotely. The attack would not leak the private information of the vehicle. And this attack would not cause fatal injuries as well because the attack is easy to be detected, and the CAV contains multi-sensors data. If the cameras break down, other sensors could still help to 'see' the environment. Based on this, the overall severity level of the attack is low.

b. Mislead camera (Fake images): With the possibility of controlling the cameras remotely, the attackers could inject fake image information to mislead the cameras. This kind of attack is more dangerous than the blind vision attack

because the detection possibility is lower. For blind vision attack, the system or the user could easily find the abnormal situation. While in the mislead camera attack, it may take longer time to detect. In addition, the system might make decisions based on the fake images, the severity of physical attack is thus higher, and the overall severity is medium.

3) Battery System: Currently, the number of electric vehicles on road is increasing. As an environment-friendly transportation method, it is believed that the future CAVs would be electric vehicles. The vehicles' battery system would then be an attack target.

The most possible attack to battery system is DoS (Denial of Services) attack. In computer science, the aim of DoS attack is to exhaust all the resources of the target to make the computer, server or communication channel unavailable. In CAVs, DoS attack could target the energy sources to exhaust the power sources, such as heating the seats on the vehicle. DoS attack could be really dangerous to the battery system. It could trigger different parts to consume battery power in a short time. Sudden battery loss could cause damage to the basic functions of the vehicle. The severity of physical damage is medium to high, and the combined severity level is high as well.

4) LiDAR (Light Detection and Ranging): LiDAR is the most fundamental sensor in CAVs and it has already been used in localisation and parking-assistance [152]. It uses light reflection point cloud to detect the distance and boundaries of surrounding obstacles and environments [153]. The importance of LiDAR is medium. There are successful attempts to attack the LiDAR [151].

a. Jamming. This attack is to jam the LiDAR by using strong lights. The attackers would use stronger light to reflect the origin light. The attackers could not gather any information through this attack. However, there are possibilities of physical damage because the detection performance of LiDAR will decrease.

b. Hidden Objects. Because LiDAR uses the reflection of light to detect the

surrounding environments, the attackers may use special materials to absorb the light to avoid detection. This attack would not cause any information leakage directly. But in some situations, if the object is covered by special reflection materials, the vehicle would not observe it, thus lead to an accident. This could cause physical damage or even fatal injuries to the target vehicle. The combined severity of this attack is thus medium.

c. Fake Objects. The attackers could use light reflection to simulate a fake object in front of the vehicle such as a barrier. The target vehicles would stop or change direction based on the false detection. If multiple vehicles detect this fake object, it could cause a severe traffic congestion. Moreover, if there are multiple fake objects on the roads, this attack could cause physical damage when CAVs try to avoid them. But as there are other detection methods on the vehicle, the possibility of causing fatal injuries of this attack exists but is low. The severity of physical damage is medium and the combined severity is medium as well.

5) Radar: Though Radar and LiDAR have similar names but Radar uses radio waves instead of light to detect the surroundings. Currently, there are two types of Radar used on CAVs, millimeter Radar [154] and Ultrasonic Radar [155]. Now, the millimeter radar is used on objects detection [156]. Ultrasonic radar is used in short distance scenarios such as parking assistance system [157]. This is because the speed of ultrasonic radar is slow, which would lead to poor detection rate in high speed movements. Radar is also of medium importance.

a. Jamming. This attack is similar to the LiDAR jamming attack. In radar jamming attack, the attackers would use noise to degrade the signal of radar. Then, the radar system might not work properly and the vehicle could not detect the surrounding environments. If the noise source influences multiple CAVs, the traffic flow would be disturbed or it could even cause traffic collisions. This attack would not cause information leakage directly, but might cause physical damage. The combined severity of this attack is medium.

b. Hidden Objects. Currently, there are existing technologies to hide objects from radar detection, which are already adapted to military aerospace area [158]. The planes or the objects would hide by changing the regular reflection shape or using radar absorbing materials. In military usage, the mitigation method is already developed, which is called Radar Anti Stealth Technology [158] to strengthen the radar signal. This attack would not cause information leakage but might cause physical damage (not directly to people). The combined severity level of this attack is medium.

c. Fake Objects. To conduct this attack, the attackers might broadcast fake radar signals. Other vehicles would then detect the false signal and conduct corresponding reactions. This attack would not cause information leakage, but might cause physical damage to infrastructures e.g. collisions when vehicles are trying to avoid fake objects. The combined severity of this attack is medium.

6) GNSS (Global Navigation Satellite System): The most well used GNSS system is GPS (Global Positioning System) from the US [159]. Currently, there are other Commonwealth or countries developing their own GNSS, such as Beidou from China, Galileo from Europe Union and Glonass from Russia [160]. GNSS system could help to locate and navigate the vehicles. Hacking into this system requires high-level knowledge. GNSS system is a major resource for positioning and navigation, but as the positioning and navigation are cooperated via V2V communication, the importance of GNSS system is medium.

a. Spoofing. GNSS spoofing is to send similar GNSS signals to target CAVs to mislead the receivers. The attackers could use these devices to perform attacks and lead the vehicle to false location or wrong route. In 2013, researchers from the University of Texas at Austin successfully fool a 80 million dollar super-yacht by their GPS spoofing devices [161]. Compared with GNSS jamming attack, GNSS spoofing attack would be more dangerous. Because without the GNSS signals, CAVs would use other methods such as V2V communication or SLAM (Simultaneous Localization and Mapping) to navigate

and avoid the possible hazards such as collisions. However if the information is wrong and not detected, CAVs would trust the wrong GNSS information and take wrong reactions, which may lead to collisions and fatal injuries. In addition, a vehicle that has been spoofed successfully could respond private information such as the location information and historic route information to the attackers, which would also cause information leakage. In that case, the severity of information leakage is medium and the severity of physical damage is high.

b. Jamming. In GNSS jamming attack, the attackers will send stronger power signal to the CAV receiver. The GNSS signal is normally weak when they approach the receivers, and it could be easily covered by the jamming signal. The real GNSS signal will then be ignored. In addition, the jamming attack is also difficult to be detected because the GNSS signal is likely to decrease due to interference or limited satellite [162]. CAVs could not navigate and locate without the GNSS signal. However, V2V communication could help to navigate coordinately as a redundancy method. According to this, the severity levels of both information leakage and physical damage are medium.

7) In-vehicle System: In-vehicle system contains the micro-controllers and devices communication instructions in the vehicle sent by CAN (Controller Area Network) or other communication methods such as WiFi, Bluetooth. The in-vehicle system is related to all the operational functions, thus is of high importance.

a. Injection. The attackers would inject non-existing information or even malware to the system through ports such as USB ports. Based on the fake information, CAVs might make wrong decisions leading to physical damage. As an active attack, injection could also cause leakage of sensitive data. The combined severity of this attack is medium.

b. Eavesdropping. Eavesdropping attack is a passive attack, which means that it is difficult to be noticed. The main target of this attack is not to cause physical damage but to gain access to valuable data. Thus, the severity of

information leakage is high and the severity of physical damage is low.

c. Traffic analysis. Traffic analysis attack is also a passive attack. The attackers will monitor and observe the data, then try to identify the pattern in the data flow. As a passive attack, traffic analysis attack would not cause physical damage directly and the information leakage scale is limited. The combined severity of this attack is low.

d. Modification. Modification attack is to modify the messages sent between different components and units. The wrong messages could lead to the wrong decision and action of the vehicle. The severity of this attack is medium.

Connectivity:

There are three main types of vehicle communication in CAV network. V2V (Vehicle-to-Vehicle) is the communication between vehicles via wireless network. V2I (Vehicle-to-Infrastructure) is the communication between vehicles and infrastructures via wireless network and V2X (Vehicle-to-Everything) contains V2V, V2I and communication between vehicles and other entities such as the cloud database or the pedestrians [163]. Compared with traditional automobiles, these communication methods could help to improve the accuracy of location in rural area and prevent accidents efficiently. Nowadays, many communication technologies are being used to fulfill the CAV network such as DSRC (Dedicated Short Range Communication), LTE (Long Term Evolution) and 5G [164].

The possible attack target assets of connectivity are listed below.

8) V2V Communication (With other vehicles): V2V communication is a crucial part in future CAVs. However, there is no general adapted communication standards for V2V communication. Currently, the V2V communication standard in USA is DSRC, which is based on IEEE 802.11p standard [165]. In Europe, there is ITS-G5 for V2V communication [166]. V2V communication could help to navigate or warn vehicles.

a. DoS. Like the DoS attacks might happen in the battery system, DoS attack could also happen in the V2V communication. The attackers could

send huge amount of data to block the communication channel of the target vehicle by wireless communication so that the target vehicle could not acquire information from outside. This attack would not cause information leakage but might cause physical damage especially in the rural area, in which the V2V communication is the main data source for vehicle planning.

b. Modification on Message/ Fake message. The communication between vehicles would send different types of information including position coordinates, speed, and head angle, etc. If the attackers send fake message, the target vehicle would take wrong reactions. In addition, if the target vehicle trusts the fake message, it may send response to the attacker, which could lead to information leakage. Based on this, the overall severity is medium.

c. Hidden vehicle. This attack is also a type of passive attack. The attackers would disable their own message sender to hide their activities. This would not cause information leakage directly, but might cause physical damage if the vehicle hide its activities and approach the target vehicle silently.

9) V2I Communication (With Infrastructure): V2I communication is the communication between CAVs and the infrastructures. Nowadays, there are some initial usage of V2I communication. For example, the ETC (Electronic Toll Collection) on roads and bridges use RFID (Radio Frequency Identification) to charge vehicles [167]. Beside the communication channel, which is similar to the V2V communication, there are other attack types in V2I communication.

a. Change infrastructure sign. The infrastructure signs are important in transportation to help vehicles to navigate, locate or control speed. CAVs could 'read' the sign and take corresponding actions. If the attackers change the infrastructure signs such as the road direction sign, it will lead the vehicle to wrong destination. In addition, if multiple traffic lights are changed intentionally, it could cause severe traffic congestion or even traffic collisions.

b. Block/ remove sign. The infrastructure signs could also be blocked or removed physically or remotely. If an emergency alert sign is removed

intentionally, this could cause traffic congestion and accidents. But this attack will not cause information leakage. The combined severity of this attack is medium.

10) V2X Communication (Mainly on Cloud)

a. Cloud ID Data set. Authority is important in CAV network. Each CAV would be assigned an unique ID such as an electronic plate. In order to confirm the reliability of the communication, only the information from the trusted CAVs in the data set could be accepted. All the communication and information exchange are based on the authority from the CAV cloud.

b. Cloud Real Time Traffic Database. Cloud database collects the traffic data to provide transportation guidance. It will include the real time traffic congestion data and accident data to inform all the CAVs to avoid relevant areas. If the attackers inject fake message or modify message, all the vehicle in the cloud database would receive wrong information. In addition, the attackers could also gather valuable information from the data set.

Based on the criteria, all the attacks are then divided into four categorises, which are shown in Table 3.4. Four attacks are ranked as critical, which are remote control, DoS attacks on battery system, GNSS spoofing attacks and authority attacks. All these four attacks could cause life-threatening consequences. In addition, except for DoS attacks on battery system, other three methods belongs to Fuzzy attack. Fuzzy attacks intend to inject modified data into communication to cause unexpected behaviours on vehicles. Thus, DoS attacks and Fuzzy attacks the most severe attacks on CAVs.

3.7 Possible Mitigation Methods

For each of the attacks mentioned above, the mitigation methods will be different. According to the mitigation methods in information security [140], the majority types of mitigation methods could be divided into five categories. To CAVs, the mitigation methods could be similar but need to be considered

Table 3.4: Attack Categories of Attack Types to CAVs

Level	Description	Attack Types
1	Critical	Remote control (Audio/Video devices); DoS attack (Battery system); Spoofing (GNSS); Fake Identity (Cloud authority)
2	Important	Mislead cameras/Fake vision (Cameras); Hidden Objects (LiDAR); Fake objects (LiDAR); Injection (In-vehicle system); Modification (In-vehicle system); Fake/ Ghost message (V2V communication); Change infrastructure sign (V2I communication); Injection (Cloud data set); Modification (Cloud data set)
3	Moderate	Blind vision (Cameras); Jamming (LiDAR); Jamming (Radar); Jamming (GNSS); Eavesdropping (In-vehicle system); Traffic analysis (In-vehicle system); DoS attack (Infrastructure sign); Block/remove sign (Infrastructure sign); Road line changing (Road)
4	Minor	Loud volume (Audio/Video devices); Fake sound (Audio/Video devices)

based on CAV characteristics.

1) Prevention: Prevention is to prevent the attacks from influencing the whole vehicle system negatively. The prevention could be achieved by encrypting the communication channel and messages, or detecting the active attacks in the systems. In addition, all the CAV users could be authorized with the credibility of the messages. For example, to the eavesdropping attacks in in-vehicle system, if the communication channel and messages are encrypted, it is much more difficult for attackers to make use of the information.

2) Reduction: Reduction is to reduce the possibility or feasibility of the attack. It could also be reducing the possible impacts of the attacks to a controllable level. In CAVs, the reduction methods include the redundancy sensors. If one sensor breaks down, the vehicle could still rely on the data from other sensors. In this case, the impact of each sensor could decrease. For example, to the blind vision attack in camera, the vehicle could then use other sensors to help after detecting abnormal attacks.

3) Transference: Transference is to share the possible risks with others, such as a reliable third-party organization. The third-party organization could be

governments and insurance companies. For example, in the Cloud of V2X communication, the authority of each CAV's identity should be assigned by the government or relevant legitimate organizations. All the CAVs information should also be stored safely and monitored by the trusted third-party. Not all the attacks could be solved through transference. In CAVs, this mitigation method could only be used when a single vehicle manufacturer or a supplier could not handle all the information safely.

4) Acceptance: Acceptance is to retain the risks caused by the attacks. This mitigation method could be used to the attacks with limited negative impacts on CAVs. On the other hand, the attack might not have a proper countermeasure and the impact is at an acceptable level. For example, to the traffic analysis attack in in-vehicle communication, the leaked information could only be the size and timing of the communication package and it is not likely to cause physical damage. In addition, the traffic analysis attack, which is a passive attack, could not be prevented by message and communication channel encryption. In that case, the traffic analysis attack could be tolerated.

5) Contingency: Contingency is to consider the possible reactions if the attacks happen. A contingency plan need to be prepared to recover the system if an attack happens. In CAVs, there could be countermeasure plans to recover the system. For example, to the DoS attack in the battery system, if the system detects an abnormal battery loss, it could pull up to a safe place.

Based on the potential attacks listed in Section 3.6, the prevention method within the framework could be achieved. It is an initial attempt to understand the vulnerabilities and risks of CAVs, the mitigation methods could be more detailed after further researches. This could help the system designers to prevent the possible attacks forehead.

However, as the CAV is a complicated system containing various sensors and ECUs, it is for sure that not all the attacks could be identified before the attacks happen. Besides, the data amount to be processed in CAVs is huge and time-consuming. It is crucial to have an automatic system to detect the

known attacks quickly and also to detect unknown attacks. This could help to reduce the possibility and severe consequences of attacks, which could be regarded as the prevention and reduction mitigation methods defined in this section. Besides, the defined potential attacks could also help designers, auto makers and suppliers to decide the severity of each attack and the obligation arrangement, which will help the transference and acceptance mitigation methods. The attacks could be mitigated by the responsible suppliers. Moreover, the attacks with low severity could be accepted. Finally, if the attacks could be detected, the response of the system and recovery are important. The consideration could be regarded as the contingency mitigation method. Based on different characteristics of attacks, appropriate reactions could be designed to response.

To develop better mitigation methods, the thesis will continue to investigate automatic anomaly detection of the attacks of the defined possible attacks within the CAVCS framework.

3.8 Summary

In this chapter, after structuring the UK CAV Cyber Security Principles, terminology, including CAVCS, CAV network, CAV vulnerability and CAV attacks, was then defined, which helps to avoid the misunderstandings in the research of CAVCS. In the UML based CAV framework, data types gathered by the CAV system were classified and vulnerabilities of CAV systems were identified. The detailed potential cyber security attacks were investigated from the aspects of target assets, risks and consequences based on vulnerabilities in the thesis.

The severity of each type of attack was then analysed based on these clearly defined criteria. Though the risks could not be quantified, it is still necessary to understand the severity of each possible attack, which could help us to prioritise attacks and provide corresponding mitigation methods. The levels of

severity for the attacks can be categorised as critical, important, moderate and minor. It was found that remote control, Denial of Service attack to the battery system, GNSS spoofing attack and attacks to cloud database authority are the most dangerous and of the highest vulnerabilities in CAVCS. Additionally, all the most severe attacks belong to DoS attack and Fuzzy attack, thus make these two kinds of attacks dangerous, which need to be solved at a high priority.

Mitigation methods including prevention, reduction, transference, acceptance and contingency were then suggested. In this thesis, a defence-in-depth attack resistant system is being built. This chapter made the first step to defining the possible attacks and severity. In the following chapters, machine learning-based anomaly detection methods will be developed and analysed on the processed data sets within the framework and evaluate for real world usage.

Chapter 4

CAV Cyber Security Data Sets

4.1 Overview

After defining the possible attacks of CAVs in Chapter 3, detecting the known attacks is the most crucial issue in building the CAV cyber security framework. However, as mentioned in Chapter 2, there exists no open CAV cyber security data set now. It is also a major research gap in the current CAVCS research.

To address this issue, several countermeasures were taken in the thesis. In this chapter, four CAVCS data sets were established, which contain simulation data and real world data, in-vehicle data and communication data. The first data set CAV-KDD, covering possible communication attacks, was generated from a computer security benchmark named KDD99; The second data set, named Simu-CAN data set, was simulated in Virtual Machine on possible attacks targeting in-vehicle communication. Two real world data sets were then introduced, KCAN-CAV data set retrieved from a car hacking data set provided by the Korean University, and real world CAV-RW data set generated from a real CAV at Wuhan University. These four data sets cover possible attacks targeting in-vehicle and communication channels. Besides, the data sets are from both simulation and real world to capture different characteristics in simulation and real scenarios.

4.2 CAV-KDD Data Set

As a fast developing topic, CAV is yet to be fully developed before they can drive safely on roads. In the existing literature on CAVs, it is difficult to access and obtain well processed and labelled data sets, especially on CAV cyber attacks. In this thesis, the widely used KDD99 benchmark data set [168] on network intrusion detection is adapted to build a CAV communication-based cyber attack data set named CAV-KDD based on the types of CAV cyber attacks and the UML based CAV framework established in Chapter 3.

This section has been published in the Journal Mathematics, titled "Machine Learning-Based Detection for Cyber Security Attacks on Connected and Autonomous Vehicles", in August 2020.

4.2.1 The KDD99 Data Set

The KDD99 data set is a well-known benchmark for online intrusion or attack detection. It was first made available at the Third International Knowledge Discovery and Data Mining Tools Competition in 1999 [168]. The KDD99 data set contains normal connection data and simulated attack or intrusion data in a military network environment. Since 1999, the data set has been the most widely used attack detection data set in the research literature [169].

KDD99 has approximately 5 million data records, each with 42 attributes (or also known as features). The 42nd attribute is the label of either normal or attack. KDD99 also provides a 10% data set with about 500 thousands data recorded for training and testing, for those who find the origin data set too big for data processing. The attacks in KDD99 are of four major types and 39 sub-attacks [170] as follows [171]:

1. PROBE, which is the Probing attack. This type of attacks monitor or scan the system vulnerabilities to gather information from the system. In KDD99, the sub-attacks of PROBE include ipsweep, mscan, nmap, portsweep, saint and satan.

2. DOS, which is the Denial of Service attack. DoS attacks disrupt the normal use or communication in the system by occupying all the resources, so that the system or communication channel are not available for normal use. Typically, the attackers would send a huge amount of data to flood the communication channel and system. In KDD99, the DoS attacks include apache2, back, land, mailbomb, Neptune, pod, processtable, smurf, teardrop, and udpstorm.

3. U2R, which is the User to Root (U2R) attack. Attackers conducting U2R attacks aim to gain access to superuser accounts. They find vulnerabilities of the system and then gain the access to the root of the system. In KDD99, the U2R attacks contain buffer_overflow, httptunnel, loadmodule, perl, ps, rootkit, sqlattack, and xterm.

4. R2L, which is Remote to Local attack. As the name indicates, the attackers aim to gain access to the system and send packets using remote connection. The attackers do not have authorized account in the system, but could gain local access to the system. In KDD99, these include ftp_write, guess_passwd, imap, multihop, named, phf, send mail, snmpgetattack, snmpguess, spy, warezclient, warezmaster, worm, xlock, and xsnoop.

It is noticeable that there are 39 sub-attacks in the four major attacks, however, only 22 sub-attacks are included in the training data set. The other 17 attacks only appear in the testing set. Testing and validation on these data sets thus also provide a measurement of robustness of detection techniques including the machine learning algorithms proposed and tested in Chapter 5.

KDD99 provides a comprehensive data set that covers a variety of attack types in computer networks. However, the data set could not be used directly for CAV cyber security due to the distinct characteristics of CAVs mentioned in Chapter 1. The thesis adapts and processes the KDD99 data set by removing irrelevant attack types based on the CAV framework established and possible attack points identified in Chapter 3. The possible attack types in KDD99 which may also happen in CAV are shown in Table 4.1.

Table 4.1: KDD99 Sub-Attacks Possibility

Attack Type		Possibility	Attack Types		Possibility
PROBE	ipsweep	H	U2R	ps	I
	mscan	P		rootkit	P
	nmap	H		sqlattack	P
	portsweep	P		xterm	I
	saint	P	R2L	ftp_write	H
	satan	P		guess_passwd	H
DOS	apache2	P		imap	I
	back	P		multihop	P
	land	P		named	P
	mailbomb	H		phf	I
	neptune	H		sendmail	P
	pod	H		snmpgetattack	P
	processtable	P		snmpguess	P
	smurf	H		spy	P
	teardrop	H		warezclient	P
	udpstorm	H		warezmaster	P
U2R	buffer_overflow	H		worm	H
	httptunnel	H		xlock	P
	loadmodule	I		xsnoop	H
	perl	I			

In Table 4.1, the possible types of CAV cyber attacks are classified into three levels, namely H for High, P for Possible and I for Irrelevant. After the data processing, the total number of CAV attack types is reduced from 39 to 14, with 19 types of possible CAV attacks and 6 types of irrelevant attack. The justifications of data processing on the attack types are listed as follows.

1. The attacks are without a clear definition. As the data are from the KDD99 data set, the definitions of attacks are referred to their original descriptions. The KDD99 data set was retrieved and processed from the DARPA intrusion detection evaluation data set collected by the MIT Lincoln Lab [172]. All the descriptions of the attacks are referenced from the official description at the MIT Lincoln Lab web site [173]. Some sub-attacks are lack of clear definitions, thus could not be classified as type Y in CAV cyber attacks. The attack type of them could be changed once a clear definition is available.

2. The attacks do not fit into the CAVCS framework. In Chapter 3, a UML based CAV structure is built to define different data in CAV communication

and functions. However, as KDD99 is a data set on computer and network security, protocols of which are different from those in CAVs. For example, in KDD99, the attack 'land' only happens in older TCP/IP protocols, and can only be found in an old Linux operating system named SunOS 4.1. Once the protocol and environment expired, the possibility of this attack may also disappear. This type of attacks does not fit into the CAV framework, thus have been removed.

3. The attacks are not compatible with the CAV potential attacks. To conduct an attack, except for the physical damage, attackers need to find one of the vulnerable points as identified in Chapter 3 in a CAV system. These attack points could be in hardware, software, data or communication channel. In KDD99, some attacks can only happen in specific conditions and platforms, thus are not applicable to the CAV attack points. The possibilities of these attacks to CAV are low. For example, attack apache2 only happens in an Apache Web Server. If a CAV does not use the Apache Web Server, the attack cannot be conducted.

4.2.2 CAV-KDD Data Set

The KDD99 data set has more than 4 million data records, and is too big for data processing on personal computers. In this thesis, the training data set with 10% selected KDD99 data is used. After removing duplicates and irrelevant attack types, a new data set which is compatible to the new CAV cyber security framework, named CAV-KDD, is established. The amount of normal data and attack data in both the training and testing data sets is presented in Tables 4.2 and 4.3.

Table 4.2: Amount of Normal and Attack Data in the 10% KDD99 and CAV-KDD Training Data Sets

	10% KDD99 Data	CAV-KDD Data
Attacks	396743	54485
Normal	97278	87832
Total	494021	142317

Table 4.3: Amount of Normal and Attack Data in the 10% KDD99 and CAV-KDD Data Sets

	10% KDD99 Test Data	CAV-KDD Test Data
Attacks	250436	23348
Normal	60593	47913
Total	311029	71261

In addition, to each sub-type attack, the amount in CAV-KDD training and testing sets are shown in Table 4.4.

Table 4.4: Amount of Sub-type Attacks in KDD99 and CAV-KDD

			10% KDD99 Training Data Set	CAV-KDD Training Data Set	10% KDD99 Testing Data Set	CAV-KDD Testing Data Set
	0	NORMAL	97278	58716	60593	47913
PROBE	1	ipsweep	1247	341	306	155
	2	nmap	231	158	84	80
DOS	3	mailbomb	/	/	5000	308
	4	neptune	107201	12281	58001	20332
	5	pod	264	40	87	45
	6	smurf	280790	199	164091	936
	7	teardrop	979	199	12	12
	8	udpstorm	/	/	2	2
U2R	9	buffer_overflow	30	5	22	22
	10	httptunnel	/	/	158	146
R2L	11	ftp_write	8	8	3	3
	12	guess_passwd	53	53	4367	1302
	13	worm	/	/	2	2
	14	xsnoop	/	/	4	4

KDD99 data set has 41 attributes and 1 label indicating the attack types. The detailed attributes could be found in Table 4.5. The 41 attributes could be classified to 4 types as follows [174]:

1. Basic connection attributes: From Attribute 1 (duration) to Attribute 9 (urgent), these attributes contain basic information about the connection such as duration, protocol types etc.

2. Connection content attributes: In the four major attack types in KDD99, DoS will show a strong time frequency change in the data flow. However, PROBE, U2R and R2L will not show the same symptom when there is an attack, the data flow will be same as the normal situation. In order to detect these kinds of attacks, the contents of the data should also be analysed, such

Table 4.5: Types of 41 Attributes in KDD99 Data Set

Type	Attributes
Basic	duration, protocol_type, service, flag, src_bytes, dst_bytes, land, wrong_fragment, urgent
Connection Content	hot, num_failed_logins, logged_in, num_compromised, root_shell, su_attempted, num_root, num_file_creations, num_shells, num_access_files, num_outbound_cmds, is_host_login, is_guest_login
Traffic Attributes based on Time	count, srv_count, serror_rate, srv_serror_rate, rerror_rate, srv_rerror_rate, same_srv_rate, diff_srv_rate, srv_diff_host_rate
Traffic Attributes based on Host	dst_host_count, dst_host_srv_count, dst_host_same_srv_rate, dst_host_diff_srv_rate, dst_host_same_src_port_rate, dst_host_srv_diff_host_rate, dst_host_serror_rate, dst_host_srv_serror_rate, dst_host_rerror_rate, dst_host_srv_rerror_rate

as the time of failed login. In KDD99, attribute 10 (hot) to attribute 22 (is_guest_login) are connection content attributes.

3. Traffic attributes based on times: Time is important in detecting attacks. For example, the DoS attack will have a much faster message frequency than normal data. Analysing attributes based on time could help to understand the connections and detect the abnormal situation. Attribute 23 (count) to attribute 31 (srv_diff_host_rate) belong to this kind of attributes. There are two types of features: "same host" and "same service", which indicate the same host or same service connection with the current connection.

4. Traffic attributes based on host: In some situation, attackers may scan ports or hosts to conduct an attack, which is known as probing. In the Probe attack, the time frequency will not be much different like the DoS attacks, so the traffic attributes based on host will be helpful to detect the attacks. It could help to find the same host connection in the past 100 connections with the current connection. Attribute 32 to attribute 41 are traffic attributes based on host.

In CAVCS, because there is no standard guidance of connections, some of the attributes for KDD99 may not be suitable for CAV attack detection. However,

all the attributes will be kept for the first data process. Because in Chapter 6, the feature selection methods will be used to find the most relevant features with the results. In case relevant features being deleted mistakenly, all the features will be used.

CAV-KDD data is then preprocessed in Weka in the following steps:

1. The normal and 14 sub-attacks are labelled as 0 to 14, as shown in Table 4.1.

2. As the data ranges of each attribute in the CAV-KDD data set and its testing set are different, some continuous data are normalized, such as duration and src_bytes. The unsupervised-attribute-normalize algorithm in Weka is used to conduct the normalization. The value range is set as 0 to 20.

3. The data then needs to be discretized. The unsupervised-attribute-discretize algorithm in Weka is used to discretize the normalized data. To other categorized attribute data such as protocol_type, service, the unsupervised-attribute-numerictonominal algorithm is used.

4. The attributes with only one value are deleted from the attribute list. These are num_outbound_cmd, and is_host_login. These attributes make no impacts on the detection as they stay the same all the time. There are, therefore, 39 attributes left in CAV-KDD.

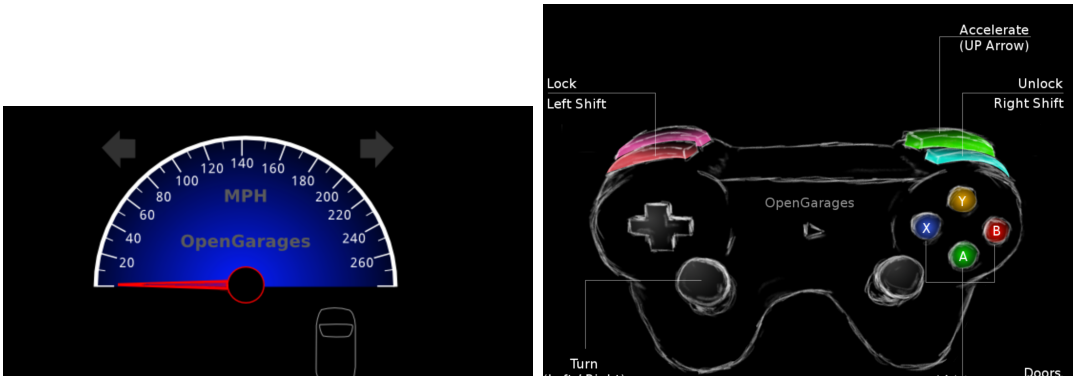
The CAV-KDD data set is more suitable for the current experiment environment. It covers the possible communication attacks, which could help to train the machine learning models to detect the attacks. This will also help the anomaly detection in V2X communication based on the universal communication features. In addition, the features will be selected and reduced based on the importance of features. This would provide guidance for real world vehicle communication feature collection. The experiments on the CAV-KDD data set will be presented in the following chapters.

4.3 Simulated CAN Data Set: Simu-CAN Data Set

CAN protocol is widely used in today's in-vehicle communication to control the response to the users' commands. This makes CAN data crucial to CAV Cyber Security, because with V2X communication in the future, attackers might take control of the vehicles remotely. However, currently, there are no fully autonomous vehicles to conduct the tests, and it is risky to conduct real world experiments directly on vehicles. It is because that firstly, real world tests might expose threats to both people and hardware. Secondly, the requirements of the tests are high to conduct a cyber attack experiment, which could only be fulfilled by a limited number of organizations and governments. Initial attempts are carried in the simulated environment of vehicles in computers. By doing this, it could reduce the risk of conducting real world experiments. In addition, the experiments conducted on computers could also be a validation process and guidance to help future real world experiments.

The in-vehicle data is collected from a CAN simulator called ICSim [175]. As the simulator could only be used in Linux operating systems, Ubuntu VMware is used to build the Linux virtual environment.

The illustration of the main interface (a vehicle dashboard) and control units are shown in Fig 4.1.



(a) *Dashboard* **(b)** *Control Unit*

Figure 4.1: *CAN Data Generator Simulator*

Figure 4.1 (a) displays the speedo meters and the vehicle body that could show the lights and doors status of the vehicle. Figure 4.1 (b) displays the control units of the simulator. In the thesis, the relevant units are controlled by corresponding keys on a keyboard, as shown in Table 4.6.

Table 4.6: Control Units of Vehicle Simulation Tools

Keyboard	Control Units
up/down	Speedo Meter
Left/Right	Lights
Right Shift+X/A/B	Open Door
Left Shift+X/A/B	Close Door

In the virtual machine, the CanDump tool is used to view and record the CAN messages in the simulation tool. With the starting instructions, the CAN simulated communication start, and messages then could be recorded as Figure 4.2.

```
(1541689200.262655)  vcan0  164  00  00  C0  1A  A8  00  00  04
(1541689200.262659)  vcan0  17C  00  00  00  00  10  00  00  21
(1541689200.262662)  vcan0  18E  00  00  6B  00  00  00  00  00
(1541689200.262666)  vcan0  1CF  80  05  00  00  00  1E  00  00
(1541689200.262669)  vcan0  1DC  02  00  00  1B  00  00  00  00
(1541689200.262673)  vcan0  320  00  00  12  00  00  00  00  00
(1541689200.264260)  vcan0  324  74  65  00  00  00  00  0E  1A
(1541689200.264285)  vcan0  37C  FD  00  FD  00  09  7F  00  1A
(1541689200.264289)  vcan0  183  00  00  00  05  00  00  10  24
(1541689200.264292)  vcan0  40C  02  36  32  32  39  53  30  39
(1541689200.264295)  vcan0  454  23  EF  36  00  00  00  00  00
(1541689200.264298)  vcan0  143  6B  6B  00  E0  00  00  00  00
(1541689200.266747)  vcan0  095  80  00  07  F4  00  00  00  17
(1541689200.266770)  vcan0  039  00  1B  00  00  00  00  00  00
```

Figure 4.2: *Simulated CAN Messages*

Based on the simulated CAN communication messages, the data could be divided into four parts. The first part is the time stamp of each message. The second part is the CAN port of the messages. In this simulated data set, only one CAN port is used, which is vcan0. This attribute is useless in the thesis

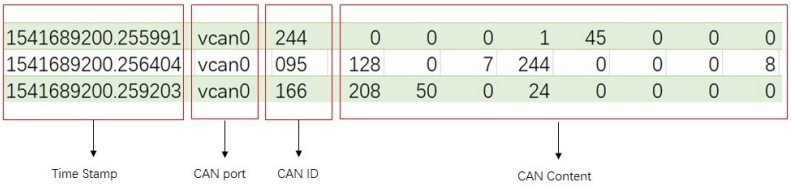


Figure 4.3: *CAN Message Format*

for detection as there is only one value. The third part is the CAN ID of each message. Different CAN IDs will correspond and control different parts of the vehicles, which will be used in the detection. The final parts of the messages are the contents of CAN messages, containing 8 bytes of CAN message.

This simulated tool is then run for three minutes to gather simulated vehicle data. The total amount of gathered data is 19493 messages. The messages are then saved and transferred into a csv file, the format of which is shown in Figure 4.3.

Because CAN messages do not contain defined attributes and labels, the features of the messages need to be defined for further analysis. The generated CAN messages are then pre-processed and labelled. The steps to define the attributes and labels are listed below:

1. Each part of the message will be regarded as one attribute in the final data set to conduct the following data processing. Because CAN port has only one value in this data collection, which is vcan0, this part will not be regarded as an attribute and will be removed from the data set. In the fourth part, the length of all the CAN contents in the collected data is 8 bytes. Then, each byte in CAN contents will be one attribute in the final data set, which becomes the attributes started from c0 to c7, indicating the different value in the CAN messages. In total, the whole data set will have 10 attributes, which are "Timestamp", "ID", "c0", "c1", "c2", "c3", "c4", "c5","c6","c7". The value of each attribute in CAN content from c0 to c7 varies.

2. In the collected data, values of "c0" to "c7" are hexadecimal, which are difficult to be processed for the machine learning models. These values are then transferred to decimal as shown in Fig 4.3.

3. All the data are then be labelled as "T", which means normal data in this data set.

These data collected from the CAN Simulator is in normal situations, indicating no attack data in the data set. Simulated attack data will then be added to the data set. As explained in Chapter 3, normally, attackers try

to modify the messages or block the communication channel. Both of these attacks could cause severe consequence such as physical damage and fatal accident. As identified in Chapter 3, the majority of the most severe attacks are related to Fuzzy attack and DoS attack. Besides, both types of attacks are active attacks, and are suitable to detect. The attack scenarios are built based on the following steps in the simulation process. Two types of attacks, namely DoS attack and Fuzzy attack, are simulated.

Attack scenario:

1. DoS Attack: Attackers inject a huge amount of data into the target CAN system in a short period. As CAN protocol is not built with any security mechanism, attackers could inject a huge amount of useless or harmful messages through unauthorised access. In addition, because CAN has only one single communication channel, the injected messages could block the whole communication channel so that the vehicle instructions could not be transmitted to relevant units on board. The vehicle would then not respond to users' commands or even respond in an opposite way. For example, the vehicle might increase the speed while the commands are reducing speed. In a dynamic environment, this could raise a life-threatening problem for the users.

In order to simulate the DoS attack, the collected data is analysed first to understand the normal pattern of the data. As the DoS attack aims to occupy the whole communication channel, it has a much quicker time frequency than normal data. In the collected data, most of the time interval between messages falls between 0.002 to 0.004 seconds. Only time intervals are changed in the DoS attacks. Simulated DoS attack data is then injected into the communication on a 0.0002 time interval. The IDs and the message are randomly generated to simulate the real world situation. In total, 5000 DoS simulated messages are injected into the data set. It should be noted that attributes c0 to c7 are not changed in DoS attacks because this type of attack would not change the value in c0 to c7. The injected attack data is labelled D, indicating the DoS attacks in the data set.

2. Fuzzy Attack: Attackers inject fake messages into the communication channel with normal time frequency. The attacker will generate messages with random ID and bytes. The attacker might target key units in the vehicles in Fuzzy attacks, such as braking or steering. Compared with DoS attacks simulated in the first scenario, Fuzzy attack aims to interrupt the basic functions of the vehicle rather than blocking the communication channel.

In order to simulate Fuzzy attacks, random attack messages are injected into the data set. In the original CAN simulation data set, most of the data time interval falls between 0.002 to 0.004 second. Based on this, in the simulated attack data, the time interval is set to 0.003 seconds. In addition, because the CAN simulated data is with different CAN ID, and has different value range of c0 to c7. For example, CAN ID 133 only has one value on c4, the value of other attributes is all 0. In this situation, the Fuzzy attack data is simulated with the following process.

a. All the data is analysed based on different CAN ID. To each CAN ID, the value range in different attributes is analysed. In the simulated attack data, the value ranges of c0 to c7 are changed.

b. Based on the value ranges of different CAN IDs, corresponding Fuzzy attack data is generated for different attributes using two types of Fuzzy attack simulation methods. The first is to generate all the random value for c0 to c7. For example, CAN ID 095 only has value for attribute c0,c2 and c3, but all the values of c0 to c7 in the attack simulations are generated randomly from 0 to 255. The second method generates specific value for specific attributes. For example, for CAN ID 133, only attribute c4 has a different value generated from 0 to 255 randomly while all the other attributes for c0 to c7 are set to 0. These simulation methods could increase detection difficulties. In addition, in real world attacks, the attackers might also analyse the data first and then conduct pertinence attacks, especially when the CAN IDs of vehicle basic functions are known.

c. The injected attack data is then labelled as F, indicating Fuzzy attack.

With these two types of attacks, the simulated normal data are extended to be the simulated CAN attack data set named Simu-CAN data set in the thesis. Thus, the amounts of different labels of the Simu-CAN data set are shown in Table 4.7.

Table 4.7: Amount of Data in Simu-CAN Data Set

Label	Number of Data
Normal data	17850
DoS attack data	4725
Fuzzy attack data	16243
Total	38818

Simu-CAN simulated the most severe potential attacks defined in Chapter 3. The data set provides the training and testing data for machine learning. The simulated data set could also be extended with new attacks if CAN is used in the vehicles. In addition, this data set also provides experiment data for further simulated attack research in real world environments. The experiments on the Simu-CAN data set will be introduced in the following chapters.

4.4 KCAN-CAV Data set

Despite the CAV-KDD data set focusing on communication, the data transmitted in the vehicles are also important. With the data exchange in V2X communication, in-vehicle data could also become vulnerable to cyber attacks, which is not the case for traditional vehicles. In Section 4.3, the Simu-CAN data set is generated and introduced to train and test machine learning models. However, real world environment generated data is of high importance, due to the reason that the simulation could not be exactly the same as real world situation, the data value might be different from the real world.

There is a lack of well-prepared open source CAVCS data sets in the literature. The CAN hacking data set named OTIDS provided by Hacking and Countermeasure Research Lab in South Korea is used in this research based on the literature review [78]. It is the most relevant open source cyber attack

data set collected from real world in the existing literature. The data set contains more than 5 million data, covering different types of attacks. Based on the OTIDS data set, a new data set named KCAN-CAV on CAN attacks to CAVs is generated and processed.

4.4.1 OTIDS Data Set

This data set focuses on the attacks on the CAN in vehicles. The four different data sets contain four types of attacks to CAN, namely DoS attacks, Fuzzy attacks, spoofing the drive gear attacks and spoofing the gauge attacks. Based on the severity assessment in Chapter 3, the most severe attacks are related to DoS and modification messages. In this thesis, the DoS attack and Fuzzy attack data set are processed to conduct further experiments.

The OTIDS data set is generated from the real world environment. The results obtained are thus more reliable and close to real world situations. A Hyundai's YF Sonata is used for generating the CAN data through the OBD-2 port of the vehicle. A Raspberry Pi3 and a laptop are also connected to acquire the data. The Simu-CAN data set and the OTIDS data sets both cover in-vehicle communications. However, the real world data in OTIDS might still be different from that of the Simu-CAN simulations. The evaluation of the machine learning models in real world automatic detection is of high importance.

Besides, the OTIDS data set has a much larger volume of data than that of the Simu-CAN data set, which would help to build and investigate more reliable machine learning models suitable for real world environments.

The data format of OTIDS is shown in Figure 4.4. The first attribute is also the time stamp of CAN messages. The second part and third part are the ID and bytes number of each CAN message, respectively. Then the next 8 attributes, c0 to c7, represent the content of CAN data. The last attribute of the data set is the label of i.e. attack or not. In OTIDS, label R represents

Column1	Column2	Column3	Column4	Column5	Column6	Column7	Column8	Column9	Column10	Column11	Column12
1478198376	0316	8	05	21	68	09	21	21	00	6f	R
1478198376	018f	8	fe	5b	00	00	00	3c	00	00	R
1478198376	0260	8	19	21	22	30	08	8e	6d	3a	R
1478198376	02a0	8	64	00	9a	1d	97	02	bd	00	R
1478198376	0329	8	40	bb	7f	14	11	20	00	14	R
1478198376	0545	8	d8	00	00	8a	00	00	00	00	R
1478198376	0002	8	00	00	00	00	00	03	0b	11	R
1478198376	0153	8	00	21	10	ff	00	ff	00	00	R
1478198376	02c0	8	14	00	00	00	00	00	00	00	R
1478198376	0130	8	08	80	00	ff	31	80	0b	7f	R
1478198376	0131	8	e5	7f	00	00	48	7f	0b	ac	R
1478198376	0140	8	00	00	00	00	08	22	2b	a3	R
1478198376	0350	8	05	20	14	68	77	00	00	2e	R
1478198376	043f	8	00	40	60	ff	7e	ce	08	00	R
1478198376	0370	8	00	20	00	00	00	00	00	00	R
1478198376	0440	8	ff	00	00	00	ff	ce	08	00	R
1478198376	0316	8	05	21	68	09	21	21	00	6f	R
1478198376	018f	8	fe	5b	00	00	00	3c	00	00	R
1478198376	0260	8	19	21	22	30	08	8e	6d	07	R

Figure 4.4: *OTIDS Data Format*

normal data while label T represents the attack data. The amount of different subsets in OTIDS are shown in Table 4.8.

Table 4.8: Amount of the OTIDS Data set

Attack Type	Total Message	Normal Messages	Injected Attack Messages
DoS Attack Data Set	3,665,771	3,078,250	587,521
Fuzzy Attack Data Set	3,838,860	3,347,013	491,847
Spoofing the drive gear Data Set	4,443,142	3,845,890	597,252
Spoofing the gauge Data Set	4,621,702	3,966,805	654,897
GIDS: Attack-free (normal) Data Set	988,987	988,872	-

In Table 4.8, the first subset of OTIDS is the DoS attack data set, which contains a large amount of CAN data with a specific CAN ID 0x000 in a short period. The time frequencies of the injected DoS attack data are every 0.3 milliseconds, which is much higher than the time frequencies of normal data.

The second subset in the OTIDS data set is the Fuzzy attack data set, with a huge amount of injected attack data with random IDs and random data value in the CAN data fields. The time frequencies of the injected Fuzzy attack data are every 0.5 milliseconds. Other two attacks, namely the spoofing the drive gear and gauge data sets, are not used in generating the new data set in the thesis.

The 4 types of attacks in OTIDS are generated and stored separately, and can be used as a binary classification problem for intrusion detection methods. However, these originally separate OTIDS sub data sets cannot be directly used for CAVs, as they still need to be pre-processed.

The research in the thesis aims to classify attacks of different categories. In addition, the whole OTIDS data set has more than 10 million data containing four data sets of different attacks and one attack free subset. Even for the DoS attacks and Fuzzy attacks, there are more than 7 million data, which is too big for a normal computer to process in this thesis or other research. Based on these factors, a new data set called KCAN-CAV is generated from the OTIDS data set and used in this research.

4.4.2 KCAN-CAV Data Set

Irrelevant and redundant data in the OTIDS data set is firstly deleted. DoS and modification are the most severe attacks based on the severity assessment in Chapter 3. Therefore, only the DoS attacks and the Fuzzy attacks in the OTIDS data set are retained to evaluate the performance of the proposed machine learning models. The data sets are then pre-processed to form a new data set called KCAN-CAV Data set.

In the processed KCAN-CAV data set, the third attribute, the bytes number of CAN messages, is removed from the original OTIDS data set as the value is same in all the CAN data. Thus, there are 10 attributes plus 1 label indicating the attack types in the KCAN-CAV data set, which is the same format as the Simu-CAN data set.

In the KCAN-CAV data set, the DoS attack and Fuzzy attack data sets are not combined because the CAN IDs are different in both attacks. The results could be unreliable if the two data sets were directly combined together. In addition, as the injected attack data has its own unique CAN ID, for example, all the DoS attacks have the same 0x000 ID. The proposed machine learning

models might classify the attacks only based on the IDs. The DoS attack and Fuzzy attack thus remain as two separate data sets. The amount of normal data and different attack data in the KCAN-CAV data set are presented in Table 4.9 and Table 4.10 respectively.

Table 4.9: Amount of Normal and DoS Attack in the KCAN-CAV Data Set

	Data amount
Normal	807,619
DoS	233,004
Total	1,040,623

Table 4.10: Amount of Normal and Fuzzy Attack in the KCAN-CAV Data Set

	Data amount
Normal	79,991
Fuzzy	3,350
Total	83,341

The attacks in the processed KCAN-CAV data set cover the most severe attack types defined in Chapter 3, and are collected and generated from real vehicles in real world environment to evaluate the machine learning models in Chapter 5.

4.5 Real World CAVCS Data Set: CAV-RW Data Set

The KCAN-CAV data set is collected from real world environment with commercial vehicles. This research explores further to collect real world data from real connected and autonomous vehicles.

Besides, the KCAN-CAV data set still has its limitations. For example, all the DoS attacks have the same CAN ID, and the attacks are generated separately, which all demand real CAVCS data of more crucial and urgent use for the literature and in this research.

Due to the lack of equipment and restrictions of COVID-19, the data collection process could not be completed at the University of Nottingham. With

the agreement of relevant research organizations, the real CAVCS data is collected from LIESMARS (State Key Laboratory of Information Engineering in Surveying, Mapping and Remote Sensing), Wuhan University.

The data collection vehicle is modified by Prof Bijun Li's research team at Wuhan University. The vehicle is a Changan model vehicle with all the essential CAV sensors installed, including Lidar, radar, GPS, etc. The photo of the experiment CAV is shown in Figure 4.5 (a). In the experiments, all the data is collected via the CANAnalyst Second Generation hardware, which is designed for CAN communication data collection. The hardware is shown in Figure 4.6. The cable is connected from the CAN-bus port on the vehicle to the computer, which is shown in Figure 4.5 (b).

(a) *CAV at Wuhan University* **(b)** *Data Collection Process*

Figure 4.5: *CAV Data Collection*

The USB-CAN Tool software is used to record the CAN communication messages. The data collection user interface is shown in Figure 4.7. As it could be seen from the user interface, the CAN message is set into the standard format with 8 bytes and in hexadecimal. The status of all the CAN messages is "received" (as shown in the 4th red column), as currently there is no injected

Figure 4.6: *CANAnalyst Second Generation Hardware*

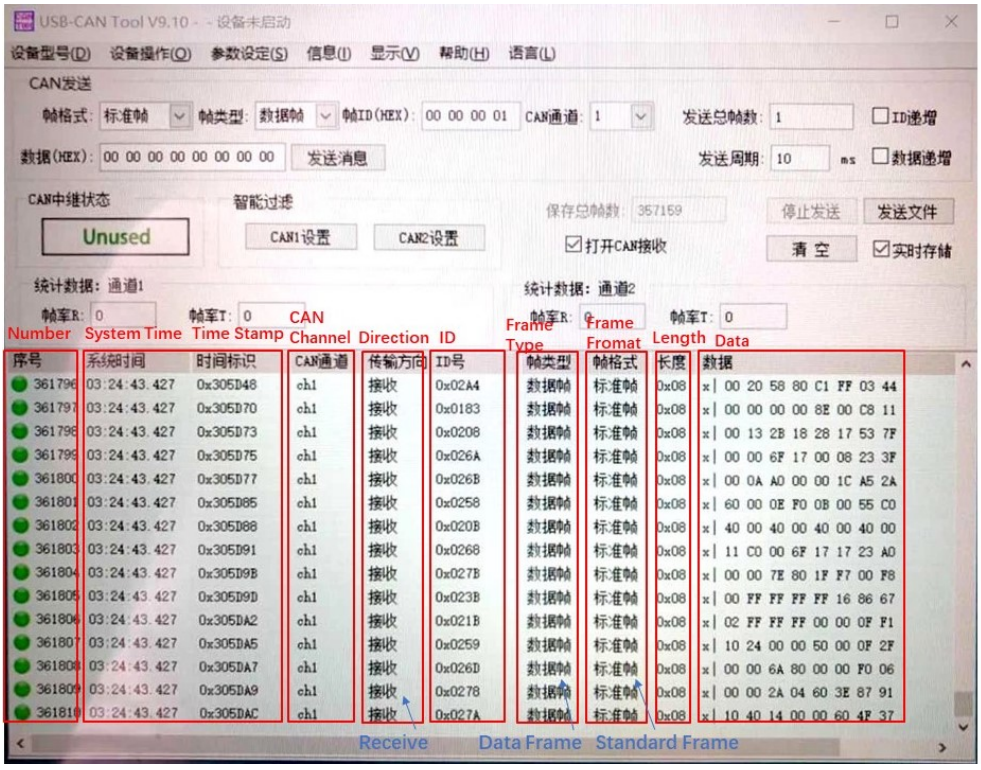


Figure 4.7: *USB CAN Tool software User Interface*

data.

In the original unprocessed data, there are 10 attributes, as shown in Figure 4.7.

The first attribute is the serial number of the received messages (as shown in the first red column). The second attribute is the system time of each CAN

message (as shown in the left of the second red column), which is the computer system time in this experiment. The third attribute is the time stamp of each CAN message (as shown in the right of the second red column), which is shown in hexadecimal. The fourth attribute is the CAN communication channel (as shown in the third red column). In this experiment, only the ch1 channel has been used. The fifth attribute is the CAN communication direction (as shown in the fourth red column). As mentioned, all the value are "received" in this experiment. The next attribute is the CAN ID of each message, which is also shown in hexadecimal. The seventh to ninth attributes (as shown in the sixth to eighth red column) show that all the CAN messages are in standard data format of 8 bytes. The last attribute is the content of the 8 bytes message.

However, not all these attributes could be used in anomaly detection, and currently, there is no attack data in this data set. The original data set needs to be pre-processed with injected attack data to generate the anomaly detection data set with the following steps:

1. Irrelevant and redundant attributes are removed from the data set. For example, the communication channel attribute only has one value, 'ch1'. This attribute value would not have any impact on the classification results.

2. All the CAN contents data are processed. In the original data set, the last attribute is the CAN content of each message. Each CAN message has 8 different bytes. These 8 bytes are split into eight different attributes. In addition, all the CAN bytes in hexadecimal are transferred into decimal.

3. The time stamps of all the CAN messages are processed. In the original data set, there are two attributes for the communication time. The first attribute is the system time of the message, which depends on the record system. This attribute is not used in the processed data set. The time stamp attribute is kept and transferred to decimal. The unit of the time stamp is milliseconds in the data.

4. All the data is then be labelled as T, indicating that they are normal data in anomaly detection.

TIME	ID	C0	C1	C2	C3	C4	C5	C6	C7	LABEL
331381	0183	0	0	0	0	0	0	0	0	T
331383	0208	0	0	0	0	0	0	0	0	T
331386	026B	0	0	0	0	0	0	0	0	T
331388	027C	0	0	0	0	0	0	4	0	T
331431	021B	18	255	255	255	0	0	5	235	T
331433	0259	16	36	0	0	80	0	5	37	T
331435	025B	65	43	2	146	0	0	0	0	T
331438	02A3	0	0	0	0	0	0	0	0	T
331440	02A4	0	0	0	0	0	0	0	0	T
331481	0183	0	0	0	0	0	0	0	0	T
331483	0208	0	0	0	0	0	0	0	0	T
331486	026A	0	0	0	0	0	0	0	0	T
331488	026B	0	0	0	0	0	0	0	0	T
331530	021B	18	255	255	255	0	0	6	234	T

Figure 4.8: *CAV-RW Data Set Format*

The processed data set now has 10 attributes in total, as shown in Fig 4.8. The first attribute is the time stamp of each attribute, the unit of which is milliseconds. The second attribute is the ID of each CAN message. The third to tenth attributes are the eight different bytes of CAN messages.

Based on the processed data set of normal CAN communications, anomalous data, which is the attack data, are injected. Same as the previous simulations, the DoS attacks and Fuzzy attacks are injected into the data set.

1. DoS Attacks. DoS attacks inject a huge amount of data into normal communication, leading to traffic jam in data exchange. This is extremely dangerous in a dynamic driving environment. The simulation of DoS attacks is the same as in the Simu-CAN data set.

Firstly, the normal data is analysed. Time frequency is the most important factors in DoS attacks, which is also the key attribute to define the potential DoS attacks. The normal time frequencies between the CAN messages are analysed firstly. It could be found that the time frequencies of messages in this communication range from 2 milliseconds to 51 seconds. The simulated DoS attack time frequency is thus set to 1 millisecond. In total, 100,000 simulated messages are injected into the original data set. The simulated DoS attacks are divided into two parts, each of 50,000 messages. Adhering to the characteristic of DoS attacks, the content of CAN messages stays the same as the original CAN messages, including the CAN ID and the byte contents in

attributes C0 to C7. The two sets of DoS attacks are injected into the data set after the 15,000th data and 200,000th data in the originally collected data set, respectively.

2. Fuzzy attacks do not modify the time frequency of CAN messages but will modify the contents of the messages. In CAV communications, the modification of message contents could be extremely dangerous because it could interfere with the operations of basic vehicle functions, including steering, accelerating or braking. The simulations of Fuzzy attacks conducted based on different contents in CAN messages.

As known from the original data set, there are 25 different CAN IDs. With different CAN IDs, the contents of CAN vary as well. Two different simulation methods are used to simulate Fuzzy attacks as follows.

a. All the contents are randomly generated in the range of 0 to 255. For some CAN IDs, different value range exists for the majority of attributes c0 to c7. For example, all the attributes of c0 to c7 for CAN ID 0208 messages are randomly generated from 0 to 255. Because in the content of CAN ID 0208, the value range of each attribute varies a lot.

b. Values of attributes from c0 to c7 for some data are generated from 0 to 255, while the rest of the attributes stay the same. According to the analysis of the original CAN data set, it could be found that only some parts of the majority of the CAN messages are changed to conduct the relevant driving tasks. To simulate the Fuzzy attacks of these messages, not all the attributes but just some parts are randomly generated. For example, all the attributes of CAN ID 0298 are 0 except for attribute c3, only c3 is randomly generated to simulate the Fuzzy attack.

It should be noticed that some CAN IDs were not simulated in the Fuzzy attacks. It is because that the original samples of these messages are too small. For example, there are only 2 records of CAN ID 0108. If more data with this ID are generated, it could then be analysed and simulated in future research.

The original data set was then processed regarding different CAN IDs. The

amount of simulated data is almost the same as that of the normal data, resulting in a data set of less bias when detecting attacks.

All the simulated data are labelled 'F', indicating the Fuzzy attacks. After injecting all the Fuzzy attacks, the size of the data set is increased to 696861 messages. With the DoS simulated attacks labelled with 'D', the total amount of normal data and attack data are increased to 796861 messages. The total amount of normal data and attack data are shown in Table 4.11.

Table 4.11: Amount of Normal and Attack Data in the CAV-RW Data set

Data	Number
Normal Data	357,159
DoS Attack	100,000
Fuzzy Attack	339,701

In the processed data set with the injected simulated data, the attributes stay the same. The first attribute is the timestamp of each CAN message. The second attribute is the CAN ID, followed by the content of CAN message c0 to c7. The last attribute is the label T, F or D indicating normal or attack data. The real world CAVCS data set is then named CAV-RW, overcoming the limitations of KCAN-CAV, which will be further analysed and processed in Chapter 5 and Chapter 6.

4.6 Summary

In this chapter, 4 different data sets covering in-vehicle and communication were introduced within the CAVCS framework, addressing the research gap of lacking open source CAVCS data sets. The detailed information of these four data sets was shown in Table 4.12.

From the table, CAV-KDD data set covers the possible communication attacks to CAVs, which was generated and processed from a widely-used computer cyber security benchmark data set KDD99.As the data set was generated and collected from network communication environment, there was no vehicle sensor used in the data set. Simulated data set Simu-CAN is then generated

Table 4.12: Summary of Four CAVCS Data Sets

Data Set	Attack Type Covered	Collection Environment	Sensors Used
CAV-KDD	Communication Attack	Communication Environment	No Vehicle Sensor Used
Simu-CAN	DoS and Fuzzy Attack	Computer Simulation Environment	Limited Traditional Vehicle Sensors
KCAN-CAV	DoS and Fuzzy Attack	Real World Vehicle Data Environment	Traditional Vehicle Sensors
CAV-RW	DoS and Fuzzy Attack	Real World CAV Environment	CAV sensors

on computer simulation tool ICSimulator. This data set covers possible DoS attacks and Fuzzy attacks to in-vehicle communications. Because the simulation tool is basic, which only covers the steering, braking and door-opening functions and relevant sensors on the vehicle.

As real world data is crucial to CAVCS researches, two real world data sets, namely KCAN-CAV and CAV-RW data sets, were introduced. KCAN-CAV data set was generated from the OTIDS data set, which is provided by the Korean University on their website. After processing, the KCAN-CAV data set covers DoS attacks and Fuzzy attacks in CAN communication.It should be noticed that KCAN-CAV data set was collected from a traditional vehicle, indicating that there is no Lidar, Radar and other CAV sensors in the vehicle. Self-collected CAV-RW data set was collected at Wuhan University by using a real CAV. The CAV-RW data set combined DoS attacks and Fuzzy attacks into one data set, making up the limitation of KCAN-CAV. In CAV-RW data set, the sensors including Lidar, Radar, Cameras and GPS receiver were used, which increase the data variety. These four data sets provided the platforms to build machine learning models to conduct anomaly detection in CAVCS. In Chapter 5 and Chapter 6, the data sets will be further discussed and analysed.

Chapter 5

Anomaly Detection based on Machine Learning

5.1 Overview

In this chapter, machine learning algorithms including Decision Tree and Naive Bayes were used to build intrusion detection machine learning models. The models were tested and evaluated on the first two simulated data sets introduced in Chapter 4, namely CAV-KDD and Simu-CAN to detect the attacks. Then, the machine learning models were adapted to the real world data sets KCAN-CAV and CAN-RW introduced in Chapter 4. Evaluation criteria including accuracy, false positive rate, model building time and testing time were analysed and compared.

The experiments have been carried out on an Intel Core i3, 3.70GHz computer with 64 bits Windows Operating System. WEKA is an open source data mining software developed by the machine learning group, at University of Waikato [176], and has been widely used in industry and research to conduct analysis and develop machine learning models.

The experiments on CAV-KDD data set in this chapter have been published in Journal Mathematics, titled "Machine Learning-Based Detection for Cyber Security Attacks on Connected and Autonomous Vehicles", in August 2020.

5.2 Machine Learning Algorithms

In WEKA, the machine learning algorithms Naive Bayes and Decision Tree were used to build the two classification models to classify and detect CAV cyber attacks.

Decision Tree is one of the mostly used classification models of a good readability [177]. It is one of the classification models structured as a tree of nodes and branches connected by one-directional edges. Each internal node of the Decision Tree (with branches leading to child nodes) represents a decision variable upon an attribute, and each branch represents a decision taken on the attribute, leading to the child nodes of different attribute values. The leaves of the tree (with no branches and child nodes) represent the classifications.

Decision Tree is selected to build the model based on its good readability, which could help to understand the priority of different attributes in the CAVCS data sets, which is useful because the research of CAVCS is still at an early stage. In addition, as currently the amount of CAV data is limited, Decision Tree could help deal with a smaller scale of data. The Decision Tree could process both numeric and nominal data, which is also helpful for CAVCS anomaly detection.

In WEKA, the Decision Tree algorithm uses the C4.5 technique to build the Decision Tree model. C4.5 conducts the classification by calculating the information gain ratio of each attribute, and chooses attributes with the biggest information gain ratio as the root node. To calculate the information gain ratio precisely, entropy carried by a data set of possible distribution values V is first calculated using Equation 5.1 as follows [178]:

$$Entropy(V) = -\sum_{i=1}^{n} p_i \cdot log(p_i) \tag{5.1}$$

Where n is the number of partitions (classification labels) of the data set, and p_i refers to the proportion of the i-th partition. Thus, the information gain could be calculated in Equation 5.2 as follows:

$$Gain(V, a) = Entropy(V) - \sum_{j=1}^{J} \frac{|V_j|}{|V|} Entropy(V_j) \tag{5.2}$$

Where a is the attribute, $|V_j|$ is the number of distributions in partition j and $|V|$ is the number of distributions in V. Thus, the information gain ratio could be calculated in Equation 5.3 as follows:

$$GainRatio(V, a) = \frac{gain(V, a)}{IV(a)} \tag{5.3}$$

In which, IV (intrinsic value) is calculated in Equation 5.4 as follows:

$$IV(a) = -\sum_{j=1}^{J} \frac{|V_j|}{|V|} log_2 \frac{|V_j|}{|V|} \tag{5.4}$$

Then, each value of the attribute will become a branch of this tree and the data could be split into different classes or tree leaves. The process will be repeated until the information gain ratio reach the threshold [179], which is set to 0.25 as default in the experiments. For example, in CAV-KDD data set, the 39 attributes are the possible distribution values. After calculating the information gain of all the attributes, the attribute dst_host_srv_serror_rate of the highest information gain is chosen to be the root node.

Naive Bayes is built based on the Bayesian probability model. Naive Bayes was selected to use because it spends less time on processing data, which fulfills the requirement of short processing time in highly dynamic driving environment. It assumes that all the attributes in the data are independent, meaning that each attribute has no impact on the other attributes [180]. Naive Bayes model calculates the conditional probabilities of classes, the class with a high probability is the prediction result [181]. The equation of Naive Bayes is presented in Equation (5) as follows [182]:

$$P(c|X) = \frac{P(X|c)P(c)}{P(X)} \tag{5.5}$$

In Equation (5), $P(c|X)$ is the posterior probability of class c when giving

predictors X. X is the data set of attributes $x_1, x_2, ..., x_n$. $P(X|c)$ is the class conditional probability of predictors X when given class c. $P(c)$ is the prior probability of class c and $P(X)$ is the prior probability of predictors X. For example, in CAV-KDD, c is the label of normal or attack data. X is the data set of 39 chosen attributes. Based on attributes of each data in the testing data set, the possibilities of them belonging to different label is calculated. Each data then is classified to the label with the highest possibility.

5.3 Evaluation Methods

Currently, there is no international standard to normalize the baseline of CAVCS performance, which adds to the difficulty of assessing the performance of the models. To better evaluate the current CAVCS models, however, criteria must be first set. This thesis will evaluate the performance of machine learning models based on accuracy, FP rate, and runtime.

Accuracy: accuracy is a widely used method for evaluating the performance of models. As shown in Equation 5.6 [183]. The N^{pred} represents all the correctly classified data. Accuracy is indicated by the percentage of correct classifications against the total data.

$$Accuracy = \frac{N^{pred}}{N^{total}} \tag{5.6}$$

FP rate: the false positive rate indicates that the data is not an attack, but the model classified it incorrectly as an attack. FP rate could thus be calculated using the following Equation 5.7 [183]:

$$FPRate = \frac{FP}{FP + TN} \tag{5.7}$$

Where FP represents the false positive, which is the data incorrectly classified as negative data. The current label data are regarded as positive data, while others are negative data. The TN represents the true negative, which is

the correctly classified negative data.

Runtime: Runtime is crucial in CAVCS evaluation due to the environment's real-time and dynamic feature. Even a one-second delay could have severe consequences. To avoid delay, the detection should be as fast as possible.

In this thesis, two kinds of runtime, namely time to build the model and time to test the model, are evaluated. The time required to build the model indicates the model's complexity. If an inordinate amount of time is needed to train a model, it might be too complicated to process, or it could be over-fitted. The time required to test the model is even more important than the time required to build it. That is because, in real time data processing, the model only needs to be built once, while it needs to be tested multiple times. To evaluate the performance of the models, both building and testing time will be considered, although the testing time of the model is of higher importance.

5.4 Machine Learning Process

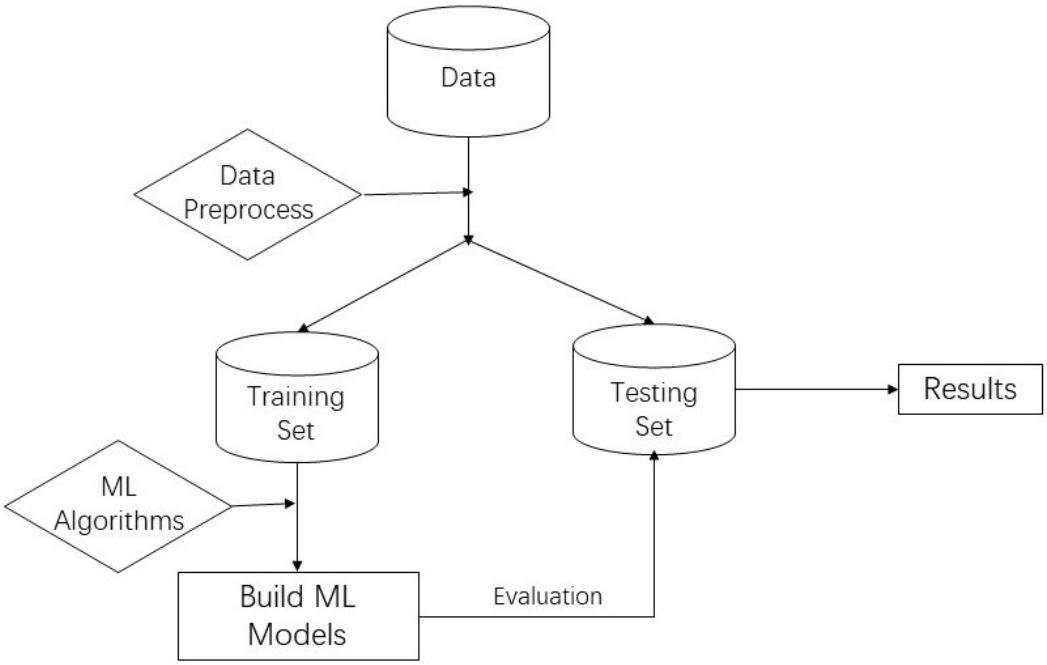


Figure 5.1: *Machine Learning Process*

The building process of machine learning models was conducted based on

the following steps, shown in Figure 5.1.

1. The data sets were first pre-processed and divided into two parts, namely training set and testing set. In the CAV-KDD data set, as there already existed a testing set, the whole original data set was considered as the training set. The other three data sets were split into 66% for training and 34% for testing.

2. The machine learning algorithms, including Decision Tree and Naive Bayes, were then adapted into the training set to build corresponding machine learning models. Two machine learning models were built in the four data sets to detect anomalies.

3. The models were then adapted to the testing sets to evaluate the performance. As discussed in Section 5.3. Multiple evaluation methods were selected to evaluate the performance of the models, and also avoid the problem of overfitting. The results will then be obtained from the performance of models on testing sets.

In the thesis, Decision Tree and Naive Bayes were used to build the machine learning models. CAV-KDD, Simu-CAN, KCAN-CAV, and CAV-RW data sets were used to build and evaluate the model. The results will be listed in the following sections. Moreover, the results will also be further improved by adapting the feature selection methods, which will be discussed in Chapter 6.

5.5 Experiments on CAV-KDD Data Set

As mentioned in Chapter 4, after processing the original KDD99 data, the number of attack types was reduced to 14 in CAV-KDD. CAV-KDD training data set was used to build the detection models, which were tested on the CAV-KDD testing data set. To avoid the overfitting problem, the training set firstly used 10-folds validation to build the model. Then the machine learning model was validated in the CAV-KDD testing data set. The overall accuracy, precision and runtime of the Decision Tree and Naive Bayes net models were compared in Table 5.1. The accuracy indicates the ratio of correct

classifications of attacks against the total amount of data.

Table 5.1: Accuracy and Runtime of the Machine Learning Model on CAV-KDD

	Accuracy on 10-folds Validation	Accuracy on the Testing Data set	Time to Build Model (s)	Time on the testing Data set (s)
Naive Bayes	99.42%	95.66%	0.15	3.38
Decision Tree	99.80%	97.04%	2.42	0.94

From Table 5.1, it can be seen that the Decision Tree model achieved the higher accuracy of the two models, while the runtime varied. In a real time driving environment, especially when CAVs are travelling at high speed, time is crucial, as a long distance of more than 30 meters can be travelled in less than a second. With almost the same accuracy, Naive Bayes needed a longer time to identify the attacks and, thus, Decision Tree was more efficient for CAV cyber security.

In addition, due to the specific characteristics of CAVs, the FP (false positive) rate of attacks classification is also a crucial metric to evaluate the performance of the models. In real world situations, if a machine learning models classifies the attack data as normal data, the consequences could be life-threatening. Based on this, the false positive rate is shown in Table 5.2. The precision of each model based on the following Equation 5.8, could also be analysed, as shown in Table 5.2.

$$Precision = \frac{TP}{TP + FP} \tag{5.8}$$

It could be seen that, with 10-folds cross validation, as all the attack types were analysed and trained, the false positive rate was much lower compared to the false positive rate on testing data set. The false positive rate of both models were similar on the testing data set and both models achieved a precision over 94% (94.84% and 94.64%, respectively). Based on these results, the false positive rate was acceptable for both models.

The accuracy and false positive rate of detecting normal and anomalous data

Table 5.2: FP Rate and Precision of Machine Learning Models on CAV-KDD

	FP on 10-folds Cross Validation	FP on the Testing Data Set	Precision on Testing Data Set
Naive Bayes	0.1%	5.2%	94.84%
Decision Tree	0.1%	5.6%	94.64%

of each sub-types of attacks is listed in Table 5.3.

Table 5.3: Accuracy of Sub-Attacks Types Obtained by Machine Learning Models

			DT Accuracy	DT FP rate	NB Accuracy	NB FP rate
	0	NORMAL	99.7%	8.3%	98.2%	7.6%
PROBE	1	ipsweep	96.1%	0%	97.4%	0%
	2	nmap	100%	0%	100%	0.1%
DOS	3	mailbomb	0%	0%	0%	0%
	4	neptune	99.1%	0.1%	97.6%	0%
	5	pod	88.9%	0%	93.3%	0.1%
	6	smurf	99.6%	0%	99.9%	0.8%
	7	teardrop	100%	0.1%	91.7%	0.1%
	8	udpstorm	0%	0%	0%	0%
U2R	9	buffer_overflow	59.1%	0%	9.1%	0.1%
	10	httptunnel	0%	0%	0%	0%
R2L	11	ftp_write	0%	0%	0%	0.3%
	12	guess_passwd	0%	0%	2.3%	0.3%
	13	worm	0%	0%	0%	0%
	14	xsnoop	0%	0%	0%	0%

From Table 5.3, it can be seen that both machine learning classification models had high accuracy when identifying CAV cyber attacks. The FP rates were low in all the attack data. When identifying the PROBE attacks, Naive Bayes performed excellently, while Decision Tree did not perform as well when detecting the ipsweep attacks. When identifying the DoS attacks, both models performed similarly; while, when detecting the pod attacks, the accuracy of Decision Tree was much higher. Both models performed poorly under the U2R and R2L attacks, due to the limited number of records of the U2R and R2L attacks in the training data sets. However, it can be seen that Naive Bayes still successfully detected 2.3% guess_passwd attacks, the accuracy of which

was slightly higher than that of the Decision Tree model.

It was noticeable that both machine learning algorithms performed poorly on attack types which were only included in the testing data set; namely mail-bomb, udpstorm, httptunnel, worm, and xsnoop. The accuracy of identifying these five attack types were all zero, meaning none of them are detected. This is due to the fact that both Decision Tree and Naive Bayes build models using supervised learning and, thus, are not able to detect unseen new attack types. Further investigations on building classification models or clustering models on unseen types of attacks remain an interesting work for future research.

Based on the results, it can be summarised that Decision Tree achieved better results, regarding to the communication-based attacks in the CAV environment. In the experiments, the Decision Tree model could detect the attack in a short time with good accuracy. However, it should also be noticed that both models obtained unsatisfactory results when predicting unseen attacks, which needs more investigations in the future studies.

5.6 Experiments on Simu-CAN Data Set

As described in Chapter 4, after simulating the attack scenarios and pre-processing, the Simu-CAN data set contains 42277 instances in total, including 17644 Fuzzy attacks and 5162 DoS attacks. The following experiments used Simu-CAN to build the detection models in CAN communication, which were tested by splitting the data set into 66% training set and 34% testing set.

The Decision Tree methods and Naive Bayes methods were used to build the detection models. The overall accuracy and runtime of these two models were compared in Table 5.4.

From Table 5.4, it could be seen that the accuracy of both models was not satisfactory. The Decision Tree could only achieve less than 95% accuracy while the Naive Bayes was even worse. Based on these results, the accuracy of each sub-attacks was also analysed, the results of which were shown in Table

Table 5.4: Accuracy and Runtime of Machine Learning Models on Simu-CAN

	Accuracy	FP Rate	Time to Build Model (s)	Time on the Testing Data Set (s)
Naive Bayes	69.7%	30.3%	0.1	0.06
Decison Tree	94.2%	5.8%	0.48	0.01

5.5.

Table 5.5: Sub-Attacks Accuracy and FP Rate on Simu-CAN

		T (Normal)	F (Fuzzy Attack)	D (DoS Attack)
Accuracy	Naive Bayes	93.4%	63.2%	0%
	Decision Tree	88.7%	98.6%	90.7%
FP Rate	Naive Bayes	49.5%	6.4%	0%
	Decision Tree	1.1%	1.4%	5.20%

It was found that the Naive Bayes failed to detect the DoS attack. In Decision Tree model, though the accuracy of detecting DoS attacks was fairly high, it could be found that the false positive rate of DoS attack detection was much higher than the other two types. It could be deduced that the DoS detection should be improved. The current machine learning models were not good enough to detect the DoS attacks.

Based on the Characteristic of DoS attack, the time is the most important attribute to detect the possible DoS attack. In the simulation of DoS attack in Simu-CAN data set, the time frequencies of each CAN message are shorten. Thus, two new attributes WithID and WithoutID were added into the data set to better detect the DoS attack.

The first attribute WithID is the time frequency between the current CAN message with last same ID CAN message. The value of WithID of first CAN message was set to 0 in the data set.

The second attribute WithoutID is the time frequency between current CAN message with last CAN message, the CAN ID could be same or different. The value of WithoutID of the first CAN message was set to 0 in the data set.

After adding these two attributes, the whole data set thus has 12 attributes illustrated in Figure 5.2. The machine learning algorithms Decision Tree and

Naive Bayes were then adapted to the data set to build the machine learning models. The results of these two models were compared and analysed in Table 5.6.

Time	ID	C0	C1	C2	C3	C4	C5	C6	C7	WithID	WithoutID	LABEL
1541689200.255990	244	0	0	0	1	45	0	0	0	0	0	T
1541689200.260510	158	0	0	0	0	0	0	0	25	0	0.00452	T
1541689200.260550	161	0	0	5	80	1	8	0	28	0	0.00004	T
1541689200.260550	191	1	0	16	161	65	0	11	0	0	0	T
1541689200.260560	133	0	0	0	0	167	0	0	0	0	0.00001	T
1541689200.262620	136	0	2	0	0	0	0	0	42	0	0.00206	T
1541689200.262640	13A	0	0	0	0	0	0	0	40	0	0.00002	T
1541689200.262650	164	0	0	192	26	168	0	0	4	0	0.00001	T
1541689200.262650	13F	0	0	0	5	0	0	0	46	0	0	T
1541689200.262650	17C	0	0	0	0	16	0	0	33	0	0	T
1541689200.262660	18E	0	0	107	0	0	0	0	0	0	0.00001	T
1541689200.262660	1CF	128	5	0	0	0	30	0	0	0	0	T
1541689200.262660	1DC	2	0	0	27	0	0	0	0	0	0	T
1541689200.262670	320	0	0	18	0	0	0	0	0	0	0.00001	T
1541689200.264260	324	116	101	0	0	0	0	14	26	0	0.00159	T

Figure 5.2: *Data Format with New Attributes of Simu-CAN*

Table 5.6: Accuracy and Runtime of Machine Learning Models with New Attributes

	Accuracy	FP Rate	Time to Build Model (s)	Time on the Testing Data Set (s)
Naive Bayes	73.5%	26.5%	0.13	0.14
Decision Tree	97.8%	2.2%	0.91	0.05

Table 5.7: Sub-Attacks Accuracy and FP Rate on Simu-CAN with New Attributes

		T (Normal)	F (Fuzzy Attack)	D (DoS Attack)
Accuracy	Naive Bayes	79.0%	64.9%	82.1%
	Decision Tree	98.2%	99.0%	92.4%
FP Rate	Naive Bayes	18.0%	5.30%	15.7%
	Decision Tree	2.4%	0.6%	0.6%

As seen from Table 5.6, the overall accuracy of Decision Tree and Naive Bayes were 97.8% and 73.5%, respectively. Compared with the accuracy without new attributes, it could be seen that the accuracy of two algorithms were both increasing. Regarding the FP rates, the FP rates were also decreasing to 2.2% and 26.5%, respectively. Both accuracy and FP rate indicated that the new attributes were useful to detect the attacks.

Besides the overall accuracy and FP rate, the performance of individual attack improved as well. Because the new attributes were more related to the

time frequency of the messages, the DoS attack detection would be affected more significantly. As seen from the results in Table 5.7, to the DoS attack detection through Naive Bayes, before adding the new attributes, the model could not detect the DoS attack, while now the detection accuracy raises to 82.1%. Though the FP rate was still not satisfactory. As for the DoS attacks detected by Decision Tree model, the accuracy increased and the FP rate decreased as well, which indicated the improvement by the new attributes. Even for the Fuzzy attack detection, the accuracy and FP rates had improvements. Based on these, the new attributes were necessary for the attack detection in CAV cyber security, and will be used in future experiments.

In addition, regarding the two machine learning models, the performance of Decision Tree was much better than that of Naive Bayes method. After adding two new attributes, the accuracy of Decision Tree model achieved 97.8% and the FP rate was 2.2%. While after adding the two new attributes, the accuracy of Naive Bayes was still low, which was only 73.5%. The FP rate of Naive Bayes was unacceptable as well.

As for the runtime, the building time of Decision Tree was 0.91s and the testing time was 0.05s. Considering that only 33% data were testing data, the testing time needs to increase as the data amount in real world would be larger. In Naive Bayes, the testing time is usually longer than building time. In this situation, the testing time of Naive Bayes was 0.14s, which was almost triple that of Decision Tree, which also indicated that Decision Tree was more appropriate for the attack detection.

Compared with CAV-KDD, the runtime of building and testing model of Simu-CAN were much quicker. It was because that the amount of data in Simu-CAN was much smaller than that of CAV-KDD. In the real driving environment, the data amount would be much larger, the limitation of the data amount could be evaluated when generating data in the real world environment, which will be further analysed in Section 5.6 and Section 5.7.

5.7 Experiments on KCAN-CAV Data Set

The machine learning algorithms have been applied to the KCAN-CAV data set. However, as the original OTIDS data set were collected separately on DoS attacks and Fuzzy attacks, the attack IDs were different and the attack were not continuous. Thus, the algorithms might detect the attack only based on the time stamps and attack IDs. For example, all the Fuzzy attacks happened after the DoS attacks. The algorithm would then classify all the abnormal data after certain time as Fuzzy attacks, which is misleading. In the experiments, the detection of attacks in the data sets were conducted separately on DoS attacks and Fuzzy attacks. Besides, the new attributes WithID and WithoutID were also injected into KCAN-CAV data set to help the detection.

The accuracy, FP rate and runtime of detection of Fuzzy attack were shown in Table 5.8. As seen from the table, the overall accuracy rates of Decision Tree model and Naive Bayes were both high, which were 99.9% and 99.4%, respectively. The accuracy of Naive Bayes was slightly lower than that of Decision Tree model.

However, it did not mean that both models achieved good results. From the FP rate results, it could be seen that the FP rate of the Decision Tree model was 1.32%, which was acceptable. However, the FP rate of Naive Bayes was high, which was 44.8%, indicating that nearly half of the data were classified incorrectly. In real world driving situation, this could lead to fatal consequences. Naive Bayes was thus not suitable for the detection of Fuzzy attack in CAVs.

Table 5.8: Accuracy and FP Rate on KCAN-CAV Fuzzy Attack

		All	T (Normal)	F (Fuzzy Attack)
Accuracy	Naive Bayes	99.4%	99.6%	55%
	Decision Tree	99.9%	99.9%	97.1%
FP Rate	Naive Bayes	44.8%	45%	0.4%
	Decision Tree	1.32%	2.9%	0%

In the previous experiments, such as the CAV-KDD, though the Naive Bayes

has a lower accuracy, the runtime was much shorter than that of Decision Tree. However, as seen from Table 5.10, the testing time on Fuzzy attacks of Naive Bayes was more than twice of the testing time of Decision Tree. Though the building time of Naive Nayes is much faster than that of Decision Tree. The quicker building time does not impact on and benefit in dynamic driving situations. The amount of the KCAN-CAV data set is less than a million, however, in real world situation, the amount might increase. As for the testing time, the Decision Tree model was still not fast enough, and needed to be improved. More detailed analysis of feature selection for improving the efficiency will be presented in Chapter 6.

Table 5.9: Accuracy and FP Rate on KCAN-CAV DoS Attack

		All	T (Normal)	D (DoS Attack)
Accuracy	Naive Bayes	97.8%	97.2%	100%
	Decision Tree	99.9%	100%	100%
FP Rate	Naive Bayes	2.2%	0%	2.8%
	Decision Tree	0.03%	0%	0%

Beside the discussed Fuzzy attacks, the accuracy, FP rate and runtime of the detection of DoS attack were shown in Table 5.9. Both machine learning models achieved good results. The accuracy of Decision Tree model was 99.9%, where only 1 message was classified incorrectly among all the 1040623 data. The FP rate was only 0.03%, as there was only one incorrectly classified data. For the sub-attacks, either normal data (T) and DoS attack data (D) both achieved fairly high results.

However, in real world, it is not easy to achieve such a good result. Because in KCAN-CAV data set, all the DoS attacks use same CAN ID, 0x000, which could mislead the machine learning models. To further analyse the potential overfitting issue in the Decision Tree model, more detailed results based on feature selection will be discussed in Chapter 6.

For Naive Bayes results on accuracy and FP rate, it could also be seen that the accuracy was 97.71% and the FP rate was 2.19%, which were both acceptable, especially compared with the performance of Naive Bayes on the

Simu-CAN data set. For each subset, the normal data had an accuracy rate of 97.2% with a FP rate of 0%, while the DoS attack had an accuracy rate of 100% with a FP rate of 2.8%. The reason that the accuracy of Naive Bayes was high was that the model classified some normal data as DoS attacks, it means some of the correct information cannot be sent to the vehicle.

Table 5.10: Runtime of Machine Learning Models on KCAN-CAV Data Set

		Building Time (s)	Testing Time (s)
DoS Attack	Naive Bayes	2.88	1.8
	Decision Tree	29.01	0.67
Fuzzy Attack	Naive Bayes	2.98	2.57
	Decision Tree	83.25	1.17

The runtime of DoS attacks by both machine learning models was shown in Table 5.10. The testing time of Naive Bayes model was slower than that of Decision Tree model, which was 1.8 seconds, while Decision Tree only used 0.67 seconds. With more than 3 million data in the testing set, the testing time of these two models was acceptable. In addition, it is also believed that the testing time could be decreased with more powerful computing facilities installed on the vehicles.

Based on all of these results, the Decision Tree is more suitable to detect the DoS attacks. Similar results were obtained in the other experiments as well. However, accuracy were too high in the results, indicating a potential overfitting issue with the Decision Tree model. This will be further analysed. The runtime could also be shortened with other techniques such as feature selection methods to remove less irrelevant attributes. This will be analysed in Chapter 6.

The separate detection of DoS attacks and Fuzzy attacks in the KCAN-CAV data set achieved a high accuracy. Based on the characteristics of the models and the data set, the reasons could be summarised as follows.

Firstly, in each subset, there was only one type of attack, either DoS attacks or Fuzzy attacks. Compared with the multiple classifications in the CAV-KDD data set, the classification in the KCAN-CAV data set is binary, which

requires simpler models. The model could also classify the normal and attack data quicker than other data set.

Secondly, the Decision Tree model achieved a better result with the well-structured attributes. In the data structure, the CAN contents, including c0 to c7, all have a certain value range. When an attack happens, the value range would change dramatically, thus can be easily detected by Decision Tree. Because in the Decision Tree model, the overall value range of 0 to 255 for tree nodes is divided for each attribute into smaller parts, the attack value could be easily localised by the model.

Some limitations of the machine learning models were also found. Firstly, the accuracy rates of both models were high. However, the Naive Bayes model incorrectly classified lots of normal data as attacks. Although the overall accuracy was high, the FP rate was high as well. The overall accuracy should not be used as the only evaluation metric in the performance evaluation. More metrics need to be introduced to build a better and more suitable anomaly detection model.

In addition, the data in the data set were not balanced. There are 807,619 normal data and 233,004 DoS attack data. The percentage of normal and attack data was acceptable for the DoS attack, as DoS attacks only cover a short period of interruptions to the communication. However, as for the Fuzzy attacks, there are 799991 normal data against only 3350 Fuzzy attacks. Due to the bias in the data set, the result of Fuzzy attack was less reliable and difficult to improve. With more Fuzzy attack data, the model could be more reliable with improved performance on detecting Fuzzy attacks.

Besides, as the two data sets were collected separately on different time periods, it was difficult to combine the two data sets. The CAN IDs of data in the data sets were different. In addition, the different periods of attack time mean that the detection of attacks can be made based on only the attack time, which will definitely obstruct the performance of the models. Because of this, the experiments were conducted separately. However, in real world

situation, it is possible that attacks are made in various and complicated way. In the experiments of KCAN-CAV data set, the detection of multiple types of attack is not conducted in this thesis. It posed a new challenge to collect the DoS and Fuzzy attack during the same time period from the same vehicles. Furthermore, the KCAN-CAV data sets were collected and simulated from a traditional vehicle, more results need to be analysed on real CAVs with self-collected CAV attack data set CAV-RW data set in the Section 5.7.

Finally, all the simulated DoS attacks used the same ID 0x000. This ID was not used in normal data. This posed a problem that the model can detect the DoS attacks based on just their CAN IDs. However, in real world situation, the attackers could use the various CAN IDs in normal data to prevent the detection. This problem will be discussed in the CAV-RW data set explained in Section 5.7.

5.8 Experiments on CAV-RW Data Set

The Decision Tree and Naive Bayes models were built to detect the DoS attack and Fuzzy attacks in the CAV-RW data set. Because there is no specific testing set, the CAV-RW data set was thus divided into two parts: 66% training data and 34% testing data.

Compared with the other real world data set KCAN-CAV, CAV-RW data set addressed several limitations. First, the data bias problem in KCAN-CAV was solved. CAV-RW then combined DoS attacks and Fuzzy attacks instead of classifying attacks separately. In addition, the only ID used in DoS was also changed in CAV-RW data set to make the attacks more similar to real world attacks.

The existing attribute Time helped to detect the DoS attacks. Like in the Simu-CAN and KCAN-CAV data set, the existing attributes were still not enough for attack detection. New attributes WithID and WithoutID were injected into the data set. The WithID indicated that time frequency between

CAN messages with same CAN ID. The WithoutID represented the time frequency between the CAN messages without considering the CAN IDs.

As mentioned in Chapter 4, there were 25 different CAN IDs in the CAV-RW data set. The data set was then split into 25 sub-data sets to analyse the time frequency of specific CAN ID. To each CAN ID, the first CAN message was marked as 0 because there was no former message. The time interval of the rest of the CAN messages were calculated, and this attribute were named 'WithID', indicating the consideration of their ID information. With the newly generated attributes, the accuracy, FP rate and runtime of each machine learning model were shown in Table 5.11 and 5.12, respectively.

Table 5.11: Accuracy and FP Rate of Machine Learning Models on CAV-RW Data Set

		All	Normal Data	Fuzzy Attack	DoS Attack
Accuracy	Decision Tree	99%	98.5%	99.6%	98.6%
	Naive Bayes	61.21%	63.1%	56.0%	72.1%
FP Rate	Decision Tree	0.4%	0.5%	0.01%	0.8%
	Naive Bayes	21.9%	32.8%	2.1%	15.7%

Table 5.12: Runtime of Machine Leaning Models on CAV-RW Data Set

	Building Time (s)	Testing Time (s)
Decision Tree	68.31	0.6
Naive Bayes	1.56	1.27

It could be seen that the Decision Tree model achieved a satisfactory accuracy with an acceptable runtime. To all the attack detection in the data set, the overall accuracy achieved 99% by Decision Tree and 61.21% by Naive Bayes, respectively. The FP rate of Decision Tree model was only 0.4%. To different sub-attacks, Decision Tree also achieved good accuracy with low FP rate. In the detection of the Fuzzy attacks and DoS attacks, the Decision Tree model achieved 99.6% and 98.6% on accuracy, with 0.01% and 0.8% on false positive rate, respectively.

However, Naive Bayes model performed poorly on both accuracy and FP rate. The overall accuracy was only 61.21%, the accuracy of Fuzzy attacks

was even lower. The FP rate of Naive Bayes was also unsatisfactory, indicating that Naive Bayes was not suitable for the Fuzzy and DoS attacks detection in real world.

Regarding the runtime including model building time and testing time, Naive Bayes model was much quicker than Decision Tree in model building time. Decision Tree used 68.31 seconds to build the model, while Naive Bayes only used 1.56 seconds. However, the building time is not as important as the testing time. Because in the real world usage, the machine learning model only needs to be built once while the testing procedure needs to be conducted several times. As seen from the table, Naive Bayes was slower than Decision Tree in testing the model on the testing set. Naive Bayes used 1.27 seconds to test on the testing data set, while the Decision Tree model only used 0.6 seconds when using the same experiment environments.

In general, the Decision Tree was much appropriate for the real world attack detection in CAVs. As it could achieve a high accuracy rate with a short testing time. It could also be confirmed that the runtime could reduce when high performance computers are used. As currently due to the experiment limits, only personal computers with limited computing power were used.

The Naive Bayes model performed poorly on the detection. This was due to the characteristics of Naive Bayes, which requires the independence between the attributes. However, in the data set, the attributes were correlated, which will have negative impacts on the performance of Naive Bayes model. From the experiment results, Naive Bayes performed poorly on all the in-vehicle data sets. This indicated that Naive Bayes is not suitable for the detection on in-vehicle data.

Decision Tree model was suitable for Fuzzy attack because the value range in all the attributes were fixed. If a Fuzzy attack happens, the data would not fall into the normal value range. The characteristics of Decision Tree could help to classify the value range quickly. In this case, Decision Tree could achieve good results in detecting Fuzzy attacks.

The experiments on CAV-RW data set mitigate the limitations of the other real world data set KCAN-CAV. First, CAV-RW data set was collected from a real CAV, indicating that the CAN data would be more various as there were more sensors and ECUS installed on the vehicle. Then, the DoS attack and Fuzzy attack were combined in the CAV-RW data set. Compared with binary classification on KCAN-CAV data set, the detection on CAV-RW data set was more complicated and more similar to real world environment.

5.9 Summary

In this Chapter, all the four CAVCS data sets, namely the CAV communication data set CAV-KDD, simulated data set Simu-CAN, real world data set KCAN-CAV and self-collected real CAV data set CAV-RW were tested by the Decision Tree and Naive Bayes machine learning models.

From the results of the experiments, several conclusions could be made. First, the machine learning models could effectively help detect attacks in the CAVCS data sets. The experiments of detection on CAVCS indicated that Decision Tree achieved better results, compared with Naive Bayes model. Regarding the accuracy in all the data sets, the Decision Tree model achieved higher accuracy rates. As for the runtime, the Decision Tree model needed longer model building time, compared with that of Naive Bayes models. However, the testing time of Decision Tree was shorter than that of Naive Bayes. As the testing time is more important than the building time, Decision Tree could achieve better results. However, the runtime of either model was still not quick enough for highly dynamic CAV driving environment. Much faster runtime is needed, and also maintaining a similar accuracy and a satisfactory FP rate.

In general, Decision Tree model achieved much better results than Naive Bayes model. This was due to the characteristics of Naive Bayes, which demands the independence between each attribute. However, in the data sets,

some attributes were correlated tightly, which will weaken the performance of Naive Bayes model. From the experiment results, Naive Bayes performed poorly on all the in-vehicle data sets, including Simu-CAN, KCAN-CAV, and CAV-RW. This indicated that Naive Bayes is not suitable for the detection on in-vehicle data. Decision Tree model was suitable for detecting attacks because the majority of the value range in the attributes were fixed, especially for the Fuzzy attack. If an attack happens, the data would not fall into the normal value range. The characteristics of Decision Tree could then help to classify the value range quickly.

Feature selection methods could help to shorten the testing time for CAVCS detection while retaining a high accuracy. They could also reduce the irrelevant attributes and increase the efficiency of the models. The improved performance of the machine learning models with feature selection will be discussed in Chapter 6.

Moreover, as discussed in this chapter, the newly added attributes WithID and WithoutID on the time frequencies between data were useful when detecting DoS attacks and Fuzzy attacks. Feature selection methods could also help to select the most relevant attributes. If the new attributes could help the model to detect attacks, the feature selection methods would also identify the high correlation between the newly added attributes and classifications by selecting them. Further improvements were needed and will be discussed in Chapter 6.

Chapter 6

Enhanced Anomaly Detection by Feature Selection

6.1 Overview

The total amount of data in the CAV-KDD, Simu-CAN, KCAN-CAV and CAV-RW data sets is more than millions. However, the amount of data is still likely to increase in real world usage. The electronic control units (ECUs) in traditional vehicles can produce 2000 CAN messages per second [22], and that number is likely to increase in well-equipped CAVs of the future. Furthermore, the channels that CAVs use to communicate with the outside environment, such as V2V and V2I, also generate large volumes of data per second. The performance of current models to deal with the large amounts of data constantly entering and exiting CAVs is still not satisfactory.

Due to these large volumes of data, machine learning models built in Chapter 5 are expected to detect the anomalies more quickly to protect CAVs in life-critical situations. Therefore, both the accuracy and runtime, as well as general efficiency, of the machine learning models should be improved to deal with greater quantities of data.

Feature selection methods in machine learning could help to address these issues by reducing irrelevant features (or also known as attributes). More-

over, understanding and choosing the most relevant features would also help real world data collection and further analysis to focus on meaningful features. This chapter proposes and analyses feature selection methods suitable for CAVCS. It also applies and compares mainstream feature selection methods in its search to find the most appropriate feature selection method for CAVCS. The most effective model and important attributes are then suggested based on the analysis.

6.2 Feature Selection Methods

Feature selection methods can be classified into three main streams, namely filter, wrapper, and embedding.

1. Filter: Filter feature selection methods do not consider the machine learning models into the evaluation [184]. They choose the feature subsets first and then adapt the machine learning models. These methods only evaluate the features based on filter parameters such as distance, information, and correlation. The filter methods generate different feature subsets based on different methods. Because machine learning models and features are independent in filter methods, time consumption under filter method will be lower [185].

2. Wrapper: Wrapper feature selection methods do consider the machine learning models as part of their evaluations. The performance of machine learning models on test sets will be evaluated to choose feature sets [184]. If the machine learning model could not be changed in some situations, wrapper methods would be a good choice to conduct the feature selection process. The feature subsets normally could lead to good results and fewer selected features because they are evaluated by specially selected machine learning models. However, when compared with filter methods, wrapper methods are more time-consuming because the machine learning models would have to be retrained several times [186].

3. Embedding: In filter and wrapper methods, the feature selection pro-

cess is independent of the training and testing of machine learning models, which means that these methods could be applied to any machine learning models if appropriate. In embedding methods, the feature selection methods are combined with the training and testing of machine learning models. Some machine learning models already have feature selection functions inside the model to conduct feature selection methods automatically [187]. However, the complexity of the model could increase. Besides, not all machine learning algorithms could use embedding feature selection methods. For example, in the Decision Tree model, the Gain Ratio has been calculated, indicating that the embedding feature selection method is already in the model.

In this thesis, filter methods (Info Gain and Gain Ratio, the Pearson Method), wrapper methods (CFS) and embedding methods (Decision Tree) were used and compared. More detailed results and discussions will be shown in Section 6.3.

6.3 Experiments

The assessment of feature selection models takes the following six steps, shown in Figure 6.1:

1. Original input features: In this step, all features in the data set will be included in the selection. For example, in the CAV-KDD data set, there are 41 features plus 1 label; In the other three data sets, including Simu-CAN, KCAN-CAV and CAV-RW, there are 10 features plus 1 label. The experiments will first generate the results with all the features.

2. Feature subset selection: All features will be considered in correlation with the classifications based on different feature selection methods. The most related features will be placed into feature subset.

3. Machine learning algorithms adaption: The chosen features will be used to re-train the machine learning models with the new selected attributes. Decision Tree and Naive Bayes will be served as the machine learning algorithms in the

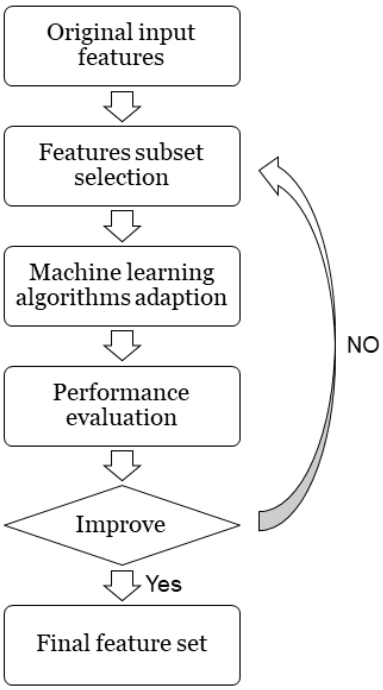


Figure 6.1: *Feature Selection Process*

feature selection process.

4. Performance evaluation: The performance of each feature selection method, including information gain, Gain Ratio, CFS (Correlation-based feature selection), and the Pearson Method, will be compared and analysed.

5. Improvement: If the results from Step 4 are improved based on the proposed feature selection method, the feature subset will be chosen as the final set to detect attacks in CAVs. Otherwise, the process return to step 2 to process again.

6. Final feature set output: The final feature set will be used to detect the attacks in CAVs and guide for real world CAV data collection. If not all the features in data set could be collected due to limited data storage and computation power, the feature set chosen by the most appropriate feature selection method should be collected first.

In the thesis, several feature selection methods were chosen to analyse and compare. The three filter methods (Pearson, Info Gain, and Gain Ratio) and the wrapper method (CFS), are chosen. The reason to choose these methods is

based on the characteristic of the machine learning models built in Chapter 5, namely Decision Tree and Naive Bayes. Decision Tree uses Information gain and Gain Ratio to select the nodes and conduct the training. Naive Bayes requires that all the features are independent, as does the CFS method. In addition, the attributes are mainly numeric. Thus, the Pearson method could help to calculate the correlation and rank the features. Based on the characteristics of these two machine learning models, the feature selection methods mentioned are chosen to conduct the process.

The concepts of Info Gain and Gain Ratio were introduced in Section 5.2. These two methods calculate and rank the Info Gain and Gain Ratio of each feature. Because of this, the Ranker search method was used in these two methods. The Ranker method will help the feature selection methods to rank the value of info gain and Gain Ratio value, which is also a default search method for these two feature selection methods.

The correlation-based feature selection (CFS) method is a wrapper method that evaluates the feature subsets using heuristics. The focus of CFS is on feature subsets rather than individual features. The heuristic method assumes that a good feature is highly correlated with the classes but not correlated with other features. This characteristic matches that of Naive Bayes, which requires the independence of each feature. The CFS equation is shown in Equation 6.1 [188]:

$$Merit_S = \frac{kr_{\bar{c}f}}{k + k(k-1)r_{\bar{f}f}} \tag{6.1}$$

Where $Merit_S$ represents the the evaluation to a feature subset S with k features. $r_{\bar{c}f}$ represents the average feature – class correlation in the subset, and $r_{\bar{f}f}$ represents the average feature – feature correlation. In Equation 6.1, it should be noticed that all the variety should be normalised, which needs to be completed in the data pre-processing step.

The CFS method uses three kinds of search methods: best first search,

forward selection, and backward elimination. To achieve better accuracy and understand the importance of each attribute, the best first search method was chosen in the experiments.

The best first search method is a type of beam search. It first calculates the feature-class and feature-feature correlation. Then, it searches all the feature subsets. It starts with an empty feature subset, then adds the highest merit feature into the subset repeatedly. If the merit value of later features is higher than before, the later feature will be kept. Otherwise, it will be removed from the subset. After all features are considered, the highest merit value feature subset is obtained [189].

Another feature selection method used in feature selection is the Pearson method. In the WEKA software, the Correlation Attribute Evaluation feature selection method is based on the Pearson correlation. Because of this, the name Pearson method was used in the experiments.

The Pearson method is a simple and easy-to-understand feature selection method. It is based on the linear relationship, and is, therefore, very fast. The Pearson method is also a filter feature selection method. The ranker method was used to evaluate the correlation of different selected attributes. The initial result of the Pearson method is in a specific value range, from -1 to 1. A value of 0 means that the correlation between the selected attribute and classification is not related linearly. A value close to 1 indicates that the attribute and the classification have a positive correlation. A Pearson value close to -1 indicates that the selected attribute and classification are negatively correlated. The Pearson method is used on the in-vehicle data sets because all the attributes in the data sets are within a specific value range, and are more likely to be linearly correlated in the classification.

The formula of the Pearson method is shown in Equation 6.2 [190]:

$$\rho(X,Y) = \frac{\sum(X-\mu_x)(Y-\mu_Y)}{\sqrt{\sum(X_i-\mu_x)^2}\sqrt{\sum(Y_i-\mu_y)^2}} \tag{6.2}$$

Where X and Y represents the two variables. X usually represents the attribute, and Y usually represents the classification, μ_x is the mean value of X and μ_y is the mean value of Y.

In CAV-KDD data set, the thesis proposed a method of gathering the intersection of attributes selected by other methods, because CAV-KDD data set has more attributes than other data sets. However, the proposed method could not be used in other data sets due to the limited number of the attributes.

6.3.1 CAV-KDD Data Set Feature Selection Results

All the features of the CAV-KDD data set, excluding the features with only one value, were used in the experiments. There are 39 attributes in total.

With different feature selection methods, different feature subsets were chosen, as shown in Table 6.1.

Table 6.1: Selected Feature Subsets on CAV-KDD

FS Method	Selected Features
CFS	service, flag, wrong_fragment, num_failed_logins, logged_in, count, same_srv_rate, dst_host_count, dst_host_srv_diff_host_rate, dst_host_srv_serror_rate
Gain Ratio	wrong_fragment, num_failed_login, dst_host_srv_serror__rate, dst_host_serror_rate, serror_rate, srv_serror_rate, same_srv_rate, flag, logged_in, urgent, count(48%)
Info Gain	service, same_srv_rate, flag, count, dst_host_same_srv_rate, dst_host_srv_rate, serror_rate, dst_host_serror_rate, dst_host_srv_serror_rate, srv_serror_rate, logged_in
Proposed Method	flag, logged_in, same_srv_rate, dst_host_srv_serror_rate

It should be noticed that only features with more than 50% correlation value in filter methods (Info Gain and Gain Ratio) were chosen to conduct the feature selection experiments. Because the CAV-KDD data set has 41 attributes in total, two of which only have one value. The remaining 39 features were sufficient to conduct feature selection experiments. When applying the Wrapper method (CFS), the number of features was set to 10 because additional features did not improve performance in this method. Additionally,

both filter methods chose 10 or 11 features, showing high correlations with the labels. As it could be seen in Table 6.1, CFS and Gain Ratio both chose 10 features, Info Gain chose 11 features. Among these, the intersection method chose four most important features, which were attribute 4(flag), attribute 12 (logged_in), attribute 27 (same_srv_rate) and 37 (dst_host_srv_serror_rate). Attribute 21 (count) also showed a high correlation with the labels. In Gain Ratio, however, the correlation value of attribute 21 was only 48%. To evaluate whether attribute 21 was important or not, the performance of accuracy, FP rate and runtime was also evaluated. It was found that after adding the attribute 21 (count), the results became worse. Therefore, the proposed intersection feature selection method did not contain attribute 21 to conduct the following experiments.

The irrelevant features were then removed from the CAV-KDD data set. The machine learning models were adapted to the CAV-KDD data set by using the chosen feature subsets. In the experiments, the thesis used the CAV-KDD training set for training and the CAV-KDD test set for testing.

Table 6.2: Accuracy, FP Rate and Runtime of Feature Selection on CAV-KDD by Decision Tree

	Accuracy	FP Rate	Building Time (s)	Testing Time (s)
CFS Method	96.6%	5.5%	0.39	0.29
Gain Ratio	95.6%	8.8%	0.43	0.23
Info Gain	96.9%	4.5%	0.27	0.57
Proposed Methods	95.9%	8.4%	0.15	0.13

Table 6.3: Accuracy, FP Rate and Runtime of Feature Selection on CAV-KDD by Naive Bayes

	Accuracy	FP Rate	Building Time (s)	Testing Time (s)
CFS Method	96.6%	5.2%	0.04	0.5
Gain Ratio	92.6%	12.2%	0.02	0.53
Info Gain	95.7%	5.1%	0.03	0.55
Proposed Methods	95.8%	8.4%	0.02	0.23

The accuracy, FP rate and runtime of the Decision Tree model and Naive Bayes model are shown in Table 6.2 and Table 6.3, respectively.

It could be seen from the tables that the runtime of the models decreased with negligible impacts on accuracy. In Decision Tree model detection, the runtime of all feature selection methods was reduced from around 1 second using all the features to less than 0.6 seconds when applying the feature selection method while the detection rate decreased within 1.5%.

In the Decision Tree model, all the feature selection methods achieved an accuracy of 95.6% or higher. Accuracy reached 97.04% when all the features in the testing set were used, as shown in Table 5.1. However, the high accuracy rate was built on high time consumption. The building time of the model was 2.42 seconds and the testing time was 0.94 seconds.

After the feature selection methods were applied, both building time and test time dropped significantly in the Decision Tree model. All the building times were less than 0.5 seconds, which decreases of at least 1.9 seconds. The testing times were all less than 0.6 seconds, showing a reduction of at least 0.37 seconds. The fastest building and testing times, 0.15 and 0.13, respectively, were both achieved by the proposed intersection method. These results showed that the building time decreased 94% and the testing time decreased 86% compared with the Decision Tree model without any feature selection method.

With the decrease in runtime, the accuracy of the detection was affected but not significantly. As it could be seen from Table 6.2, after applying the feature selection methods, all the accuracy rates were still above 95%. The lowest accuracy rate was 95.6% with Gain Ratio, and the highest accuracy rate was achieved by Info Gain, which was 96.9%. The proposed intersection method achieved 95.9%, a decrease of 1.24% compared with the Decision Tree model without any feature selection method. Based on the results, Decision Tree model with feature selection achieved improved performance.

The results of the Naive Bayes model were shown in Table 6.3. The building time of this model also decreased by at least 0.11 seconds. Considering the consumption limitation of the experiment platform, it could be deduced that if the performance of the processors on CAVs were higher, the building time

would be negligible. The fastest building time is 0.02 using the Gain Ratio and the proposed method. The testing time of the Naive Bayes model with feature selection methods reduced as well. After applying feature selection models, the testing time decreased from 3.38 seconds to at least 0.55 seconds, which means that the results improved a lot. The proposed method achieved the fastest testing time, which was 0.23 second.

Regarding the accuracy of Naive Bayes with feature selection methods, which was also shown in Table 6.3, it could be seen that the accuracy still decreased, but on a smaller scale. Compared with the accuracy of 95.66% using all the features in Table 5.1, all the other feature selection methods, excluding Gain Ratio, still maintained an accuracy higher than 95%. The proposed method achieved a high accuracy rate of 95.8%. The worst accuracy result was from the Gain Ratio method, which was only 92.6%. The accuracy of other feature selection methods increased as well.

After analysis of the results from the tables, the accuracy of both Decision Tree and Naive Bayes with feature selection methods achieved acceptable rates of accuracy. It could be seen that the accuracy of the Decision Tree was more stable than that of Naive Bayes, as the accuracy of all the feature selection methods was higher than 95%. In conclusion, it could be seen that except for Naive Bayes with Gain Ratio, all other feature selection methods could achieved acceptable accuracy rates.

In addition to accuracy, the other performance evaluation criterion, the runtime, was satisfactory as well. Before applying the feature selection methods, the building time of Naive Bayes was much faster than that of Decision Tree. After using feature selection methods, however, though the building time of Decision Tree was still higher than that of Naive Bayes, the testing time of Decision Tree was only half of that of Naive Bayes. In building time, the quickest results of Decision Tree and Naive Bayes were 0.15 and 0.02, respectively. In testing time, the fastest results of Decision Tree and Naive Bayes were 0.13 and 0.23, both of which were achieved by the proposed method.

As mentioned in the performance evaluation in Chapter 5, testing time is more important than building time. This is because the machine learning models only need to be trained once, while they need to be tested every time to detect attacks. The Decision Tree with feature selection method was more suitable for attack detection. The Naive Bayes model showed poor performance in testing time. That is mainly because that Naive Bayes has a limitation that it assumes that all the features are independent in a data set. When the features are correlated, the efficiency of the model decreases instead. The results of the experiments also showed this. After applying feature selection methods with a decreasing number of features, the correlation between features increased instead. Besides, in the real world environment, it is not realistic that all data is independent, which would have negative impacts on performance. Though Naive Bayes is quick, it is still not the most appropriate model with feature selection applied to attack detection on CAVs.

The performance of the Decision Tree model was improved by the feature selection methods. It could be seen that the runtime decreased 94% and 86% in building and testing time with a decreased accuracy rate of only 1.5% or less. It could be deduced that with the feature selection methods, the performance of the Decision Tree model could be improved, and the accuracy could be more stable in all methods, which means that the Decision Tree could be more suitable for feature selection.

The proposed intersection feature selection method achieved good performance in both Decision Tree and Naive Bayes. The decreased of accuracy was within 1.16%, but the runtime decreased significantly both in building and testing time. It could be said that the proposed method had good performance in attack detection. The proposed method could shorten the runtime with limited impacts on accuracy. It should also be noticed that the runtime is based on the experiment platform when using feature selection methods. If CAVs with high-performance processors could train and test the model in a shorter period in the real world environment, the runtime could be further shortened.

All the results still need to be evaluated in the real world environment.

In CAV-KDD data set, the best performance model is the Decision Tree with the proposed intersection model. Based on the model, it could also be deduced that the most important attributes are the selected attributes by the proposed intersection method. The four attributes, including flag, logged_in, same_srv_rate, dst_host_srv_serror_rate, indicate the connection and login status, and also the same service in the communication.

6.3.2 Simu-CAN Feature Selection Results

The feature selection methods were also applied to the Simu-CAN data set based on the steps mentioned in Section 6.2. After applying different feature selection methods, different feature subsets were chosen, which are listed in Table 6.4.

Table 6.4: Selected Feature Subsets on Simu-CAN

FS Method	Selected Feature
CFS	Time,C2,C7,WithID,WithoutID
Gain Ratio	WithID,Time,WithoutID,C5
Info Gain -30%	WithID,Time,WithoutID,C7,C3
Info Gain - 50%	WithID,Time,WithoutID
Pearson	C7, Time, C5

Like the CAV-KDD data set, the CFS, Gain Ratio and Info Gain methods as well as the Pearson method were chosen to compare the results. It was found that the Gain Ratio had poor selection results. The correlations of the attributes were low. In addition, when applying the Info Gain method, two correlation values, 50% and 30%, were chosen because there are only three attribute correlation values that are over 50%.

As it could be seen from Table 6.5 and Table 6.6, for overall accuracy, the feature selection did not achieve good results. The accuracy of the Naive Bayes models was lower than the original results. The Decision Tree models achieved even lower accuracy after applying feature selection methods. Only the accuracy of the CFS method and Info Gain method with 30% correlation

Table 6.5: Accuracy and FP Rate of Feature Selection on Simu-CAN by Decision Tree

	FS Method	All	True	Fuzzy	DoS
Accuracy	CFS	95.5%	97.1%	94.3%	93.5%
	Gain Ratio	88.8%	77.9%	99.6%	93.3%
	Info Gain-30%	96.2%	96.0%	97.3%	93.4%
	Info Gain-50%	88.1%	76.9%	98.7%	94.2%
	Pearson	85.8%	82.8%	85.1%	99.9%
FP Rate	CFS	4.5%	5.8%	1.1%	0.8%
	Gain Ratio	11.2%	1.8%	16.4%	0.8%
	Info Gain-30%	3.5%	3.5%	2.2%	0.7%
	Info Gain-50%	11.9%	2.3%	17.1%	0.9%
	Pearson	8.5%	11.6%	6.1%	0.5%

Table 6.6: Accuracy and FP Rate of Feature Selection on Simu-CAN by Naive Bayes

	FS Method	All	True	Fuzzy	DoS
Accuracy	CFS	65.5%	59.3%	65.7%	88.5%
	Gain Ratio	66.0%	92.5%	33.5%	77.6%
	Info Gain-30%	71.6%	75.6%	65.4%	77.9%
	Info Gain-50%	60.0%	83.2%	28.5%	80.9%
	Pearson	71.3%	95.9%	64.2%	0%
FP Rate	CFS	14.2%	12.9%	13.3%	22.5%
	Gain Ratio	23.8%	47%	2.7%	8.1%
	Info Gain-30%	28.4%	20.2%	14.7%	10.3%
	Info Gain-50%	40.0%	37.8%	7.8%	17.2%
	Pearson	28.8%	49.1%	0.4%	0%

value decreased slightly, from 97.8% to 95.5% and 96.2%, respectively. The accuracy of the other method was less than 90%, which dropped significantly compared with the original accuracy.

The FP rates increased as well, indicating that the amount of incorrectly classified data was also growing. The CFS method and the Info Gain method with 30% correlation value achieved the lowest FP rates within all the feature selection methods at 4.5% and 3.5%, respectively, while others were all around 10% by Decision Tree.

As for the sub-classifications in the data set of Decision Tree model, the results also varied. For detecting the normal data, only the accuracy rates of the CFS method and Info Gain method with 30% correlation were acceptable. Though some feature selection methods, including Gain Ratio and Info Gain

with 50%, achieved a lower FP rate, their accuracy rates were low. For example, the FP rate of the Gain Ratio method was only 1.8%, which was the lowest among all the normal data detection rates. However, the accuracy was only 77.9%, indicating that this is not a good model to use. In detecting Fuzzy attack, all the accuracy was satisfactory except that of the Pearson method, which was only 85.1%. The FP rates of the CFS method and Info Gain with a 30% correlation value, were both low, which were 1.1% and 2.2%, respectively. However, the accuracy of CFS was 94.3%, which was lower than that of Info Gain.

In detecting the DoS attacks by Decision Tree, as opposed to the Fuzzy attacks, the Pearson feature selection method achieved the best accuracy of 99.9% with the lowest FP rate of 0.5%. Compared with the original accuracy of detecting DoS attacks (92.4%), the accuracy of all models increased slightly after applying feature selection methods. It is also noticeable that the FP rates of models only increased slightly, which means that the models could classify the DoS attack data precisely.

After applying the feature selection methods to Naive Bayes model, it could be seen that the results were not good. Although the original results were insufficient as well (73.5%). All accuracy rates were just over 60%, lower than the original one. The accuracy rates of the Pearson method and the Info Gain method with 30% correlation value were not much affected, which were 71.3% and 71.6%, respectively. The accuracy of the rest of the models was under 70%, which was too low to detect the attack.

Among all the models, after feature selection methods were used, the CFS achieved the lowest FP rate at 14.20%, while the highest FP rate was 39.98% when applying the Info Gain method with a 50% correlation value. The two most accurate methods, namely the Pearson method and Info Gain with a 30% correlation value, only achieved 28.75% and 28.40%, respectively, on FP rates, which were all slightly higher than the original rates. It could be seen from these results that the feature selection methods did not help to improve the

overall performance of the Naive Bayes model.

Each sub-classification, including normal data, Fuzzy attack and DoS attack, also showed various performance levels. For normal data, two feature selection methods, namely the Gain Ratio method and the Pearson method, achieved high accuracy of 92.5% and 95.9%, respectively. However, the FP rates of these two models were also the highest, which were 47% and 49.1%, respectively. These high FP rates indicate that the models classified massive amounts of data incorrectly, which would be dangerous in a real world situation. Because of this, the high accuracy rates of the models are useless in real world applications. Only the FP rate of the CFS method was lower than the original one (18.0%). However the accuracy was only 59.3%, which is not acceptable for attack detection. Based on this, all the methods performed poorly on detecting the normal data. In the previous experiments on the Decision Tree model, the accuracy and FP rate of normal data were crucial to the overall performance of different models.

For Fuzzy attack data, all the models performed poorly on accuracy, especially the Gain Ratio method and the Info Gain method with 50% correlation value, the accuracy of which was only 33.5% and 28.5%, respectively. The other three models achieved acceptable accuracy, two of which had even higher accuracy than that of the original model (64.9%). These are the CFS method (65.70%) and the Info Gain method with a 30% correlation value (64.20%). However, it could also be seen that the FP rates of these two models were the highest, which indicates that the performance was not acceptable. The poor accuracy suggests that none of the models could detect the Fuzzy attack accurately.

But after applying the Pearson method, the FP rate of detecting Fuzzy attack was only 0.4%, and the accuracy was 64.2%, which was only a slight decrease compared with the original one. This indicates that the feature subset selected by the Pearson method is highly correlated with the classification of Fuzzy attack. Based on this, feature subsets were investigated, and the Pearson

method selected attributes that including c7, Time and c5. However, as it could be known from the characteristics of Fuzzy attacks, the time frequency did not change in the whole data transmission. Thus, the attributes of c7 and c5 are key features to detect the Fuzzy attack. After analysing the entire data set, it was found that in most of the data, the data value changed in c7 and c5 while the rest of the CAN content value normally stayed the same. As this attack was simulated from a CAN simulator, this still needs to be further investigated in real driving data.

For the DoS attack in the Simu-CAN data set, the accuracy is much higher than that of the Fuzzy attack. However, it was interesting that the Pearson method could not detect the DoS attack completely. Compared with other feature subsets, the Pearson method did not select related features of the DoS attack, while all the other feature selection methods selected WithID or WithoutID. The missing key features made the detection impossible.

Among all the models, the CFS method achieved the highest accuracy in detecting DoS attack, which is 88.5%, even higher than the original one (82.1%). After analysing the features selected by the CFS method, it was found that the CFS method was the only method that chose WithID and WithoutID. As DoS attack is highly correlated with time frequencies, these two new added attributes could improve the performance. However, the FP rate of the CFS method, which was 22.5%, was not satisfactory. The high FP rate indicated that the method classified normal or Fuzzy attack data into the DoS attack, which is still not adequate for the classification. Other two models achieved accuracy of 77.6% and 77.9%, respectively. The corresponding FP rates were 8.1% and 10.3%. These two models achieved similar accuracy and FP rates.

Based on all the results, the performance of detection on DoS attack is better than on Fuzzy attacks. However, as the poor performance of Naive Bayes, the accuracy of these models was not as high as that of Decision Tree, which indicates that Decision Tree is more satisfactory for detection on Simu-CAN data set.

In the CAV-KDD data set, the runtime of Decision Tree and Naive Bayes had a distinct time difference. However, in the Simu-CAN data set, though the building time of the models still varied, the testing time showed that Naive Bayes was slower than the Decision Tree, which is same as that of the CAV-KDD data set.

Table 6.7: Runtime of Feature Selection on Simu-CAN

Model	FS Method	Building Time (s)	Testing Time (s)
Decision Tree	CFS	0.72	0.01
	Gain Ratio	0.45	0.01
	Info Gain-30%	0.64	0.02
	Info Gain-50%	0.31	0.01
	Pearson	0.33	0.01
Naive Bayes	CFS	0.04	0.05
	Gain Ratio	0.03	0.04
	Info Gain-30%	0.04	0.05
	Info Gain-50%	0.03	0.04
	Pearson	0.02	0.02

After applying the feature selection methods, the runtime dropped significantly in detection by the Decision Tree model. As seen in Table 6.7, all the models achieved 0.01 second except for the Info Gain method with a 30% correlation value, the testing time of which was 0.02 second. It should also be noted that the minimal time unit in WEKA is 0.01, which indicates that the testing time could have been even quicker. However, the data amount in the Simu-CAN data set was not large, leading to a short testing time.

The building time also decreased, though it did not affect the performance of the models. Among all the feature selection applied to the Decision Tree model, the Info Gain method achieved the best result. The accuracy was the highest with the lowest FP rate, and the testing time was acceptable as well. However, compared to the original model, the results were not good enough, as the testing time was only a little faster while the accuracy and FP rates were affected.

In the experiments of the Naive Bayes model, the testing time also decreased, but not as much as that of the Decision Tree model. The Pearson method had

the fastest testing time. However, as discussed in the previous analysis, the Pearson method performed poorly on DoS attack detection, which is completely unacceptable in real usage. The rest of the models all took 0.04 or 0.05 seconds, slower than the 0.01 second achieved by Decision Tree models. Based on the runtime, Decision Tree is still more suitable for CAV cyber security detection than the Naive Bayes method.

After analysing the results, it could be concluded that in the Simu-CAN data set, the Decision Tree model was more suitable than the Naive Bayes model, no matter whether it was evaluated by accuracy, FP rate, or runtime. This is because that the attributes in the data set have a specific data range, especially the data in the CAN field. As the Decision Tree model classifies the data based on different value ranges to build the tree, the performance of the Decision Tree model was good in attack detection.

However, the feature selection methods did not improve the performance of the models. First of all, the number of attributes in this data set is not as large as that of the CAV-KDD data set. The number of attributes is 39 in CAV-KDD, while that of the Simu-CAN data set is only 10. The fewer attributes indicate that the relationship between attributes and classification is more closed, which means that if the attributes are deleted, the results would be more affected.

In addition, it was also found that every method had chosen the attribute 'WithID', indicating that this attribute is highly correlated with the classification label. Because DoS attack is related to the time frequencies of the data, the attribute WithID helped better recognise the DoS attacks. In some situations, the time frequencies of data changed to send multiple commands simultaneously. This was probably not a DoS attack but might be incorrectly recognized as such if only the time stamp of each data was used. The new attribute WithID helped to prevent this incorrect classification in certain circumstances. In this experiment on detecting DoS attack, after removing less relevant features such as the CAN content, which is useless in DoS attacks, the

model recognised the DoS attacks more accurately. That is why the accuracy rates in detecting DoS attacks rise after applying feature selection.

6.3.3 KCAN-CAV Feature Selection Results

Based on the previous results, feature selection methods need to be applied to improve the performance of the models and overcome the limitations mentioned in Chapter 5. As the DoS attack and Fuzzy attack were detected separately in the detection process, the feature selection methods were also applied separately. Several feature selection methods, including the CFS method, Pearson method, Gain Ratio and Info Gain method, were adapted to the experiments. Different feature subsets were selected by different feature selection methods. For each feature selection method, the full training data set was used to conduct the experiments, and the attribute 'Label' was set to be the target label. The chosen feature subsets are listed in Table 6.8 and Table 6.9 for DoS attack and Fuzzy attack, respectively.

Table 6.8: Selected Feature on KCAN-CAV DoS Attack

FS Method	Selected Attributes
CFS	C0,C1,C3,C5,C6,C7,WithID
Pearson-20%	C5,C1,C3,C7,ID,C0,C2,C4,C6
Pearson-30%	C0,C2,C4,C6
Gain Ratio	C5,C0,C1,C4,C3
Info Gain-20%	ID,WithID,Time,C5,C0,WithoutID,C1,C4,C3
Info Gain-30%	ID,WithID,Time

Table 6.9: Selected Feature on KCAN-CAV Fuzzy Attack

FS Method	Selected Attributes
CFS	C1,C3,C6,WithID
Pearson	WithID,C6,C2,C1

It should be noticed that when applying the Pearson method and Info Gain method, different correlation values were chosen with which to compare the results. Correlation values of 30% and 20% were chosen in both the Pearson and the Info Gain methods for comparison. As it could be seen from Table

6.10 and Table 6.11, the feature selection methods are effective in detecting DoS attack.

Table 6.10: Accuracy and FP Rate of Feature Selection on KCAN-CAV DoS Attack by Decision Tree

		All	Normal Data	DoS Attack
Accuracy	CFS	99.9%	100%	100%
	Pearson-20%	99.9%	97.2%	100%
	Pearson-30%	100%	100%	100%
	Gain Ratio	97.23%	96.4%	100%
	Info Gain-20%	100%	100%	100%
	Info Gain-30%	100%	100%	100%
FP Rate	CFS	0%	0%	0%
	Pearson-20%	0%	0%	0%
	Pearson-30%	0%	0%	0%
	Gain Ratio	0.8%	0%	3.6%
	Info Gain-20%	0%	0%	0%
	Info Gain-30%	0%	0%	0%

Table 6.11: Accuracy and FP Rate of Feature Selection on KCAN-CAV DoS Attack by Naive Bayes

		All	Normal Data	DoS Attack
Accuracy	CFS	97.8%	97.2%	100%
	Pearson-20%	97.8%	97.2%	100%
	Pearson-30%	91.3%	88.8%	100%
	Gain Ratio	96.7%	95.8%	100%
	Info Gain-20%	96.8%	95.8%	100%
	Info Gain-30%	100%	100%	100%
FP Rate	CFS	0.6%	0%	2.8%
	Pearson-20%	0.6%	0%	2.8%
	Pearson-30%	2.5%	0%	11.2%
	Gain Ratio	0.9%	0%	4.2%
	Info Gain-20%	0.9%	0%	4.2%
	Info Gain-30%	0%	0%	0%

Table 6.12: Accuracy and FP Rate of Feature Selection on KCAN-CAV Fuzzy Attack by Decision Tree

		All	Normal Data	Fuzzy Attack
Accuracy	CFS	99.9%	100%	98.3%
	Pearson	99.9%	99.9%	97.6%
FP Rate	CFS	1.7%	1.7%	0%
	Pearson	2.4%	2.4%	0%

When applying feature selection methods on DoS attack by Decision Tree models in Table 6.10, the models all achieved high accuracy except the Gain

Table 6.13: Accuracy and FP Rate of Feature Selection on KCAN-CAV Fuzzy Attack by Naive Bayes

		All	Normal Data	Fuzzy Attack
Accuracy	CFS	99.6%	99.9%	31.6%
	Pearson	99.6%	99.9%	37.1%
FP Rate	CFS	68.2%	68.4%	0.1%
	Pearson	62.7%	62.9%	0.1%

Table 6.14: Runtime of Feature Selection on KCAN-CAV DoS Attack

Model	FS Method	Building Time (s)	Testing Time (s)
Decision Tree	CFS	23.33	0.41
	Pearson-20%	10.19	0.28
	Pearson-30%	2.95	0.25
	Gain Ratio	6.51	0.26
	Info Gain-20%	23.39	0.27
	Info Gain-30%	5.07	0.19
Naive Bayes	CFS	1.73	1
	Pearson-20%	2.11	1.3
	Pearson-30%	1.15	0.87
	Gain Ratio	1.37	0.76
	Info Gain-20%	2.02	1.1
	Info Gain-30%	0.61	0.58

Table 6.15: Runtime of Feature Selection on KCAN-CAV Fuzzy Attack

Model	FS Method	Building Time (s)	Testing Time (s)
Decision Tree	CFS	26.8	0.21
	Pearson	42	0.49
Naive Bayes	CFS	0.88	0.66
	Pearson	1.01	0.75

Ratio method, the accuracy of which was 97.23% with an FP rate of 0.8%. The accuracy of the CFS method and the Pearson method with 20% correlation was 99.9%, and the rest of the models achieved 100%. However, the accuracy of 100% was not precise enough for analysing the results, as the amount of data was huge. Some incorrectly classified data was ignored when calculating the accuracy percentages. Based on this reason, the amounts of incorrectly classified data were listed in Table 6.16. As it could be seen, though the accuracy of each method was nearly the same, the numbers of incorrectly classified data were not the same. Except for the Gain Ratio, the numbers are all low.

Table 6.16: Number of Incorrectly Classified Data on KCAN-CAV DoS Attack

FS Method	Number of Incorrectly Classifications
CFS	68
Pearson-20%	1
Pearson-30%	0
Gain Ratio	9772
Info Gain-20%	1
Info Gain-30%	0

In addition, as it could be seen from the Table 6.10, the FP rates after applying all the methods were fairly low, all the methods except for the Gain Ratio method achieved an FP rate of 0%. Combining the FP rate and accuracy results, though the performance was improved, the results were unrealistic in the feature selection process. After analysing the selected features, it could be found that if the feature selection method chose the attribute 'ID', the accuracy and FP rate became extremely high. Only 1 or even 0 data were classified incorrectly in the data set. As discussed in Chapter 4, all the DoS attack data use the same CAN ID '0x000' in the data set. The models were built to be based solely on the CAN ID attribute.

In order to further analyse this, only the attribute ID was kept in the data set, and the accuracy was still 100% with an FP rate of 0%, which indicates that the feature selection methods are not suitable for this data set. Even if the model could achieve fairly high accuracy with a low FP rate, it still could not be used in real world attack detection. The same situation happened to Naive Bayes model as well, which made the results useless in the experiments.

Regarding the feature selection results of the Fuzzy attack seen in Table 6.9, the Gain Ratio and Info Gain methods were removed from the experiments because after conducting the feature selection process, it was found that all the correlation values between the attributes and the classifications were under 10%. This indicated that the feature selection results would not be good. Only CFS and Pearson methods were used in the Fuzzy attack data set. It should also be noticed that when applying the Pearson method, the correlation value

was set to be 10% as only the correlation value of the attribute 'WithID' was over 30%. If there was only one attribute left in the data set, though the runtime would be shortened, the accuracy would not be acceptable.

As it could be seen from Table 6.12, compared with the previous experiments, the overall accuracy of the Decision Tree models achieved 99.9%, though both the overall FP rates increased as well. However, with regard to accuracy and FP rate of normal and Fuzzy attack data, the accuracy of detecting Fuzzy attack increased to 97.6% and 98.3%, respectively, and both saw a decrease in FP rate to 0%.

The testing time dropped significantly after applying the feature selection methods shown in Table 6.15. The testing time of the Pearson method by Decision Tree was only half the original one and the testing time of CFS is even lower, which was only 0.21.

When applying the feature selection methods on Fuzzy attacks by Naive Bayes models, the accuracy of both models also increased, as shown in Table 6.13. However, FP rates of Naive Bayes models were too high, at 68.2% and 62.7%, respectively. It was found that the reason that FP rates were high was because that the model classified a great deal of normal data into attack data, which means that feature selection methods were not suitable for the Naive Bayes model.

The testing time of the Naive Bayes models was not good enough compared with that of the Decision Tree model. CFS achieves a testing time of 0.66 seconds, which is three times as much as that of Decision Tree models. The Pearson method is even slower than that of CFS.

Several conclusions have been made after analysing the results of the feature selection methods adapted to the KCAN-CAV data set.

First, the feature selection methods are not suitable for DoS attack detection in the KCAN-CAV data set. This is because all the attacks use the same ID, which means that all the attacks could be easily classified using one single ID value. However, in a real world situation, the DoS attack could be more

deceitful with multiple IDs and other attributes. Thus, the accuracy of detecting DoS attacks is too high, which is caused by over-fitting. It could be deduced that this model could not be applied to other data set if the attack ID changed. In order to solve this problem and better test the reliability of the machine learning model, more complex and detailed DoS attack data needs to be collected and analysed.

Second, in detecting Fuzzy attacks, the CFS achieved better results than the Pearson method. CFS method achieved higher accuracy with a very short testing time. However, the same problem with the DoS attack also needs to be considered here. The format of Fuzzy attack data in the KCAN-CAV data set is almost the same. Moreover, the amount of Fuzzy attack data is not sufficient to build a well-performing model. More Fuzzy attacks need to be added to the data set to build a more comprehensive and reliable model.

Third, it was also found in the experiments that not all the feature selection methods could achieve better results in this data set. The number of the attributes in KCAN-CAV is not as high as that of the CAV-KDD data set, almost all the attributes were important. Meanwhile, the Naive Bayes method achieved poor results in the majority of the methods. This is because the attributes are not independent, which goes against the characteristics of the Naive Bayes. Because of this, the Naive Bayes results were even worse than the former results.

Based on these results and limitations, it was found that the current open source CAV attack data set was not enough for anomaly detection. The limitations need to be addressed.

6.3.4 CAV-RW Feature Selection Results

Since the number of attributes in the CAV-RW data set is more than ten, and not all the attributes are correlated tightly with the normal and attack classifications, the feature selection process was also conducted aiming to improve

the performance of these two models.

Several feature selection methods were chosen to conduct the experiments, including CFS, Pearson method, Info Gain, Gain Ratio and Decision Tree. The reason of choosing these methods is because that they are all related to the characteristics of the machine learning models. For example, the Info Gain and Gain Ratio are important parameters in the Decision Tree model. In addition, the Decision Tree algorithm is also a feature selection method. During the model building period, the most important features were selected by the Decision Tree model. The CFS and the Pearson methods are related to the Naive Bayes models. It should also be noted that the CFS used the best first search method. Decision Tree did not use search methods while the others used Ranker search methods. The selected features of the methods were listed in Table 6.17. These feature selection method cover the three mainstream feature selection methods: filter, wrapper and embedding.

Table 6.17: Selected Feature Subesets on CAV-RW

FS Method	Selected Attributes
CFS	Time,c0,c1,c5
Decision Tree	WithoutID,WithID,c5,c2
Pearson-20%	c6,c5,c0,c2,c1,c4
Pearson-30%	c6,c5,c0
Info Gain-30%	c5,Time,c2,c1,c6,c0
Info Gain-40%	c5,Time,c2,c1

All the results for accuracy, FP rate, and runtime after applying these feature selection methods are shown in Table 6.18.

Two feature subsets were chosen by the Pearson feature selection method and Info Gain method, while other feature selection methods chose only one feature subset. In the Info Gain method, when the Info Gain value was set to be over 40%, only attribute 'c5' and attribute 'time' were selected. In order to get more reliable and comprehensive results, the Info Gain value was set to 30% to gather another subset of features. Two feature subsets were chosen by the Pearson method as well, the correlation values of which were over 30%

and 20%, respectively.

Table 6.18: Accuracy and FP Rate of Feature Selection on CAV-RW by Decision Tree

		All	Normal Data	Fuzzy Attack	DoS Attack
Accuracy	CFS	94.70%	95.2%	92.6%	99.9%
	Decision Tree	94.60%	95.6%	94%	92.8%
	Pearson-20%	86.66%	98.7%	98.4%	3.3%
	Pearson-30%	85.55%	99.5%	95.8%	0.3%
	Info Gain-30%	98.2%	97.5%	98.5%	99.9%
	Info Gain-40%	92.1%	93.3%	88.5%	99.9%
FP Rate	CFS	3.3%	5.3%	1.5%	1.7%
	Decison Tree	2.1%	5.5%	1.8%	1.6%
	Pearson-20%	10.6%	23.2%	0.2%	0.5%
	Pearson-30%	11.8%	25.9%	0.3%	0.1%
	Info Gain-30%	0.7%	1.1%	0.01%	1.3%
	Info Gain-40%	5.3%	8.4%	3.1%	1.7%

Table 6.19: Runtime of Feature Selection on CAV-RW by Decision Tree

	Building Time (s)	Testing Time (s)
CFS	40.1	0.49
Decision Tree	60.14	0.48
Pearson-20%	37.65	0.36
Pearson-30%	20.42	0.34
Info Gain-30%	31.65	0.37
Info Gain-40%	22	0.45

Table 6.20: Accuracy and FP Rate of Feature Selection on CAV-RW by Naive Bayes

		All	Normal Data	Fuzzy Attack	DoS Attack
Accuracy	CFS	58.3%	89.1%	42.9%	0.1%
	Decision Tree	47.2%	51.5%	30.1%	90.4%
	Pearson-20%	67.6%	84.0%	70.3%	0%
	Pearson-30%	65.2%	81.4%	67.1%	0%
	Info Gain-30%	69.5%	78.8%	70.4%	32.9%
	Info Gain-40%	63.9%	78.2%	67.6%	0%
FP Rate	CFS	33.5%	64.8%	10.3%	0.1%
	Decison Tree	23.8%	34.3%	10.4%	31.8%
	Pearson-20%	25.7%	42.7%	15.4%	0%
	Pearson-30%	27.7%	44.7%	17.8%	0%
	Info Gain-30%	22.7%	34.8%	15.7%	2.6%
	Info Gain-40%	28.1%	43.5%	19.9%	0%

The highest accuracy rate of Decision Tree was achieved by the Info Gain with a 30% correlation value, which is 98.2%, with 31.65 seconds building

Table 6.21: Runtime of Feature Selection on CAV-RW by Naive Bayes

	Building Time (s)	Testing Time (s)
CFS	1.23	1.13
Decision Tree	0.97	0.93
Pearson-20%	1.35	1.03
Pearson-30%	0.75	0.69
Info Gain-30%	1.26	0.94
Info Gain-40%	0.82	0.63

time and 0.37 second testing time. The lowest accuracy rate was achieved by Pearson method with a 30% correlation value, which was 85.55%, with a 20.42 second model building time and a 0.34 second testing time. Though the testing time is low, and it is almost half the testing time before feature selection, the accuracy is too insufficient to conduct the detection. It should also be noted that the Pearson methods with a 20% correlation value showed poor accuracy, which was only 86.66%. The FP rate of the Pearson method also increased from the original 0.4% to 10.6% and 11.8%, respectively. These all indicated that the Pearson feature selection method is not appropriate for the Decision Tree model.

In addition to each sub-attack, it could be seen from the table that the Pearson method showed poor accuracy in the detection of DoS attacks. The Pearson's accuracy is only 3.3% on 20% correlation and 0.3% on 30% correlation, respectively. After analysis and comparison with other feature selection methods, it could be found that the Pearson method was the only method that does not select attributes: timestamp, WithID and WithoutID. As the DoS attack is highly related to the time frequency between messages, this feature selection led to a bad result, indicating that the Pearson feature selection method is not suitable for anomaly detection.

All the feature selection methods chose attribute c5, and the majority of the methods chose c0, c1, c2, c6, and timestamp. This indicates that in this collected data set, most CAN message values were changed in these attributes. However, only the Decision Tree chose WithID and WithoutID together. In

addition, from the tree structure built by Decision Tree, these two attributes were the most related attributes to the classifications.

Normally, the feature selection methods could help to decrease the runtime without decreasing the accuracy significantly. However, as shown in Table 6.18, all the feature selection methods show a decreasing trend in accuracy. Even the highest accuracy achieved, by the Info Gain method, was only 98.2%.

The runtime of models after applying each method was quicker than in the original model, as shown in Table 6.19. But the decreases in runtime were more significant on the model building time than testing time. As said in the performance evaluation, model building time is not as important as the testing time, indicating that the feature selection methods are not good enough to improve the performance.

As for the FP rate, it could be seen from Table 5.11 that before feature selection, the FP rates in general and for sub-attacks were low enough, which are all below 1%. After applying feature selection methods, almost all the FP rates increased greatly, some of which are even beyond 20%. This indicates that massive data were classified falsely, which might cause severe consequences in the CAV dynamic driving environment. Only the Info Gain with a 30% correlation value achieved an FP rate of 0.7%. Regarding the accuracy, building time and testing time, the performance of Info Gain with a 30% correlation value is the most satisfactory.

Based on all these criteria, it was found that for real world CAV data, feature selection would not provide more efficiency or improvement to the original model. This is because all the CAN fields from c0 to c7 are important in making driving decisions. Hence, any missing attribute could cause the wrong classification of the data. In addition, time is an indispensable attribute in detecting DoS attacks. It could be seen from the Pearson method, without any time attribute such as WithID or WithoutID, that the detection accuracy of DoS is terrible. Even worse, no DoS attack were detected without time attributes in Pearson with a 30% correlation value method. Based on this, it

was found that for real world CAN data, the integrity of the data is important. Only through contents attributes and time attributes could the attacks be detected. It should also be noted that the Decision Tree achieved a high accuracy rate, but its testing time is still not good enough to be applied to CAVs. More efficient methods to improve the time consuming is of high importance, which need to be further investigated.

6.3.5 Discussions

According to all the experiments on the four data sets, the performance of accuracy, FP rates and runtime were compared. Thus based on different attacks in the four data sets, the best performance model and important attributes are listed in Table 6.22.

Table 6.22: Best Models and Important Attributes of Feature Selection Results

Data Set	the Best Performance Model	Important Attributes
CAV-KDD	Proposed intersection method	flag, logged_in, same_srv_rate,dst_host_srv_serror_rate
Simu-CAN	Decision Tree model with Info Gain	DoS Attacks: Time, WithID, WithoutID; Fuzzy Attacks: c7,c5
KCAN-CAV	Decisio Tree model with CFS	DoS Attack: ID,WithID, Time; Fuzzy Attacks: c1,c6,WithID
CAV-RW	Decision Tree with Info Gain	DoS Attack: Time,c5,WithID; Fuzzy Attacks:c5,c2,c6,c0

The best models were suggested based on four criteria: high overall accuracy, high sub-attacks accuracy, low FP rate and fast runtime. Among all the results, the accuracy needs to be above at least 95% to be regarded as a high accuracy rate.

After analysing all the results, the performance of Decision Tree is superior than that of Naive bayes. In all the four data sets, Decision Tree model achieved better performance than Naive Bayes, especially Decision Tree model with Info Gain.

It could also be seen from the results that the feature selection methods

achieved good results on CAV-KDD data set. It is because that CAV-KDD data set has 39 attributes in total, while other data set only has 10 attributes. The feature selection method could achieve better results in data set with a large amount of attributes.

For communication attacks, the important features are related to the connection and login status, and the percentage of same service. It indicates that in future communication data collection, these attributes need to be considered at high priority. For DoS attacks, the time critical attributes are crucial, including Time, WithID and WithoutID. The newly added attributes were found useful on the detection.

In addition, the detection of Fuzzy attacks depends on the CAN field contents from c0 to c7. When detecting the attacks, it could be found that the attributes including c0, c2, c5 and c6 are important in CAV-RW. While in other data sets, different attributes are important. It is because that the sensors used in the data collection is different, and the data contents changed as well. However, this will also raise another issue, the data bias could affect the results. For example, if all the data only change one value in the CAN field, such as c0, the importance of c0 will become extremely high. The content meaning in CAN needs to be further investigated in future research to resolve the data bias problem.

6.4 Summary

In this chapter, feature selection methods, including CFS, Info Gain, Gain Ratio, the Pearson method, and others were adapted to machine learning models built in Chapter 5. Several findings could be summarised as below.

First, feature selection methods have been identified to be effective in the improvement of models performance. Especially on the CAV-KDD data set, the runtime decreased with little impact on accuracy. However, the feature selection methods do not perform well on the following three data sets: Simu-

CAN, KCAN-CAV and CAV-RW, due to the limited number of features in these three data sets. It is found that the feature selection method is more efficient in data sets with large number of attributes.

Secondly, feature selection methods are more effective when applied to the Decision Tree models. In the experiments, the accuracy of the Decision Tree model remained stable while the accuracy of Naive Bayes varies. It is found that the Decision Tree model with feature selection methods is more appropriate for CAVCS attack detection.

In addition, newly added attributes "WithID" and "WithoutID" have been identified useful for the DoS attacks classifications. The majority of feature selection methods selected the new attributes.

Meanwhile, the proposed intersection feature selection method can only be used when there are sufficient number of attributes, such as in the CAV-KDD data set. Otherwise, the number of selected attributes will be limited, which will lead to an unsatisfactory model performance.

Besides, experiments of feature selection also provided the understanding of the most relevant and important features when detecting different attacks in simulated or real world data. For communication attacks in the CAV-KDD data set, the attributes flag, logged_in, same_srv_rate and dst_host_srv_serror_rate, covering connection and login status, and same services percentage, are the key features. Time-related features 'Timestamp', 'WithID' and 'WithoutID' were considered to be the most important features in detection of DoS attacks. The CAN content attributes c0 to c7 are also vital parts when detecting fuzzy attacks. The mentioned features need to be collected first in case the computation power and storage for data are limited on CAVs, which will ensure that the performance of model are not affected significantly. In addition, the selected key features will also serve as guidance for future research.

In conclusion, feature selection methods on CAVCS detection should be further analysed. More feature selection methods need to be considered and adapted to machine learning models. Each feature in CAN data also needs to

be further investigated regarding different functions of CAVs. The experiments in this chapter provide an initial attempt to acquire higher accuracy with a short runtime. The newly added features could also serve as guidance for further CAVCS data collection.

Chapter 7

Conclusions

7.1 Main Contributions

CAVs are becoming the most interesting research topic around the world. However, there still exists a large number of unsolved problems. Being a prevalent part of CAVs safety, CAVCS needs to be addressed properly. Any method or solution that helps to enhance the CAVCS could also become a contribution to the development of CAVs.

In the thesis, a CAVCS framework was built based on existing standards and principles, including UK CAV cyber security principles. The main aim and contribution of the thesis is to provide a machine learning-based anomaly detection cyber security framework for CAVs. Several other contributions have also been made in the thesis during the building of the framework.

The existing literature, including CAV standards, commercial and technical reports, relevant journals and conference papers, were reviewed and presented in Chapter 2. The unique CAVCS characteristics have been discussed and compared with traditional network and vehicles. Methods of building a cyber security framework, including risk assessment, related data sets, anomaly detection, and feature selection, were also reviewed.

Based on the literature, the CAV cyber security terminology and definitions are defined to set the theoretical foundation of the thesis in Chapter 3, followed

by a clearly defined list of vulnerabilities and potential attacks to CAVCS. This potential attack list could be extended in the future easily as the technologies of CAVs are still fast evolving. In addition, a new severity assessment method for CAV attacks was proposed in Chapter 3 as well. The severity assessment method evaluated different attack severity based on well-defined criteria, including risks, assets and consequences. It was found that the DoS attack and Fuzzy attack are the most dangerous threat to CAVs. The severity assessment method could also be adapted to new attacks to assess the severity, which could help to control the attacks better and make appropriate responses.

Four new data sets were introduced in Chapter 4, including CAV communication data set CAV-KDD data set, in-vehicle simulated communication data set Simu-CAN data set, real world data set KCAN-CAV and CAV-RW data sets. The four data sets cover both the in-vehicle and inter-vehicle data to help build and evaluate machine learning models. The data sets were all retrieved or collected by myself, also contributing to CAVCS research as currently there is no open source processed CAVCS data sets.

Among all the mitigation methods suggested in Chapter 3, the detection of attacks in CAVs is the most important, which is also the aim of the thesis. In Chapter 5, machine learning-based anomaly detection in CAVs was proposed. Two machine learning models, namely Decision Tree and Naive Bayes, were evaluated on the newly-collected data sets.

The performance of two machine learning models was then compared. It was found that the machine learning models could help to detect the attacks in CAVs. Besides, the Decision Tree could achieve much better results than Naive Bayes in all the data sets, especially the in-vehicle data sets. However, the performance of which could still be improved through feature selection methods.

In Chapter 6, main stream feature selection methods were used and compared. With little impacts on the accuracy, the runtime of each model decreased significantly. The best model and important features were suggested

in this chapter. However, it was found that the feature selection methods only had limited effects on improving the performance on data sets with fewer attributes, which indicated that the correlation between attributes and classifications are tighter as well. At the same time, after applying the feature selection methods, Decision Tree could achieve quicker testing time, even quicker than that of Naive Bayes. Thus, Decision Tree model is more appropriate for CAV cyber security attack detection. The outputs of this research could be listed as below:

1. The definitions of CAVCS and potential cyber attacks to CAVs. This definition could be used in future CAVCS research. Meanwhile, the potential attack list could be adapted to different vehicle models to find their potential attack points. In addition, the attack list is also extendable to add new types of attacks.

2. A new severity assessment method for potential cyber attacks. The severity assessment method used a well-adapted engineering formula to assess the cyber attack from risk, asset and consequence. The criteria could be used for newly defined attacks, helping the researchers or engineers to rank the priorities of different attacks and mitigate them.

3. New CAVCS data sets for CAVCS research. This thesis retrieved CAV communication data set CAV-KDD from the benchmark KDD99 by deleting irrelevant and redundant data. Meanwhile, the thesis also simulated the Simu-CAN data set on computers. Real world data set KCAN-CAV was also retrieved from the OTIDS data set from the Korean University. In addition, a completely new data set CAV-RW was collected from real world CAVs. All of these data sets could be reused to conduct other machine learning-based CAVCS research.

4. The building and comparison between machine learning models, including Decision Tree and Naive Bayes, provided solutions and suggestions for CAV attack detection. Machine learning algorithms were used to build models and the models were then compared based on accuracy, FP rate and runtime. It

was found that the Decision Tree could achieve better results.

5. The performance improvement of machine learning models by feature selection and newly added features. Several feature selection methods,including filter, wrapper and embedding, were used to find the relevance of each attribute. The best performance model towards different attacks in the four data sets were suggested. Moreover, the selected important attributes could provide guidance for future CAVCS data collection. In addition, the added attributes WithID and WithoutID have been considered to be effective in CAVCS anomaly detection, especially on DoS attack.

Based on the outputs above, a machine learning-based anomaly detection CAV cyber security framework was then established, covering potential attack assessment, CAVCS data sets, attack detection, and model improvement, which could be used to conduct further CAVCS research.

7.2 Limitations and Suggestions for Further Improvements

Gathering real world CAV anomalous data used in anomaly detection is one of the biggest challenges faced during this research. Currently, there are very few real CAVs that are open to the public, not to mention the CAV cyber attack data. Although the data sets used in the thesis contained real world data collected from real CAVs, only normal data could be collected, the attack data in the data sets were still simulated. It is difficult to obtain real world attack data due to safety and privacy problems. In order to overcome this limitation, specific testing equipment and testing fields need to be used to gather real 'real world' data. The different levels of automation and connections, and different attack scenarios also need to be considered when collecting data in the future.

In addition, data bias still exists in the data sets. For example, in the CAV-KDD data sets, several types of attacks only have few data while other types

have thousands. In in-vehicle communication data sets, the CAN fields have data bias, so that the performance of feature selection methods could not be guaranteed. For example, in Fuzzy attack data of the KCAN-CAV data set, the majority data only changed the data value on attribute 'c6'. Thus, the importance of c6 became high in the feature selection process. However, the situation could be different in other real world data collections. The meaning of data in CAN fields needs to be studied comprehensively to understand the activities of the vehicles. Moreover, the evaluation methods of attacks need to be developed. In the thesis, the attacks were leveled into different categories. However, it is more useful if the risk of attacks could be quantified.

Another limitation is the performance of the computer used in the experiments. All the computers used in this research are still personal computer with limited computing power, which will have negative impacts on the performance of the model. If high performance computers are used to process the data, the results could be improved on runtime. Moreover, currently there is no conclusion on which kind of CPU will be installed on CAVs. If the data could be processed in real CAV CPUs, the results could also be more representative and reliable.

Besides, the thesis was also affected by the impacts of the COVID-19, which limits the field tests in more scenarios.

7.3 Future Work

As initial attempts to raise the awareness of CAVCS and also attempts to apply machine learning models to detect the attacks in CAVs, two peer-reviewed journal papers and one conference paper have been published. Two more publications are also prepared to be submitted to journals. In addition, all the data sets retrieved and collected in the thesis were publicly online so that other researchers could also use the data sets to conduct other researches related to CAVCS, which could enhance the impact of this research. The following

research topics are the potential research areas for future research.

1. Comprehensive Data Sets

Currently, there is no open source CAVCS data set available online. All the data sets in this research were retrieved and collected independently. Though the data sets mentioned in the thesis have been published online, it is still not enough for CAVCS research. First of all, capable organizations or companies could help to conduct real attacks to driving CAVs (in a controlled testing field). Moreover, experts on cyber security, CAVs and other relevant fields could help to define more potential attacks and collect them. Thus, the attack data sets could be more comprehensive including more types of attacks and real driving situation attack data rather than simulation data. Besides, data from different levels of CAVs also needs to be considered to better understand the CAVs.

2. Model Performance Improvement

In this research, only supervised machine learning algorithms are used to conduct the detection. However, the supervised machine learning algorithms showed poor performance on unseen attacks. On the one hand, more machine learning algorithms could be used. The performance could then be analysed and discussed, which could help to find the most appropriate model for different attacks. On the other hand, more types of algorithms, such as unsupervised machine learning algorithms, could be used to increase the detection rate of unseen attacks.

3. Real-time Anomaly Detection

In the thesis, all the data sets were collected and pre-processed first, and then classified. All the data in the data sets are well-formatted and could be used to conduct the experiments directly. However, in a real driving situation, the data could be in a mess and cannot be used directly. Appropriate methods need to be developed to parse the data once received. Based on these methods, the detection could be conducted immediately on the vehicle, which will significantly improve the efficiency of anomaly detection and enhance the

security of the CAVs.

4. Other Mitigation Methods

As mentioned in Chapter 3, several mitigation methods could be used to enhance the safety and security of CAVs. In this research, only the detection methods were investigated. Other mitigation methods, such as authentication and encryption of messages, could also be used to improve the difficulty of attacking a CAV. Especially for passive attacks, authentication and encryption are more useful than detection. Though on current CAVs, limitations such as the consuming capabilities and limited bytes of transferred message might hinder mitigation research. It still would be a promising research direction to find universal mitigation methods for a defence-in-depth CAVCS framework.

7.4 Summary

This work provided a holistic framework for Connected and Autonomous Vehicles cyber security research from definition, detection to mitigation. Relevant terminology was proposed in Chapter 3. The definition of CAVCS was proposed first to build the theoretical foundation of the whole research. Then, potential attacks were listed and severity assessment criteria were built to rank the priority of different attacks. The severity assessment criteria were based on the risk, asset and consequence of different attacks, and each attack has been marked as low, medium and high as the severity level. It was found that DoS attacks and Fuzzy attacks are the most dangerous and of the highest vulnerabilities in CAVCS. Based on the attacks, several mitigation methods were also suggested, the most efficient of which is machine learning-based anomaly detection.

In the existing literature, machine learning algorithms were found powerful to build models to detect the attacks in CAVs. Decision Tree and Naive Bayes machine learning algorithms were chosen in this research to build models to detect the attack. Meanwhile, new data sets, including communication data

sets CAV-KDD, simulated in-vehicle data set Simu-CAN, real world vehicle data set KCAN-CAV and self-collected real world CAV data set CAV-RW, were collected and investigated in this research. All the data sets are now open online for other researchers to use.

Machine learning models were applied and compared on these CAVCS data sets. It was found that the Decision Tree model could achieve much better results than Naive Bayes. In order to improve the performance of models, feature selection methods including CFS, Gain Ratio, Info Gain and Pearson method were applied to find the most relevant attribute in the data sets. For data set with massive attributes, feature selection methods could play an important role in improving the performance. However, for in-vehicle data sets, such as Simu-CAN, the effect of the feature selection method was not significant, which was attributed to the smaller number of attributes. New added attributes WithID and WithoutID, indicating the time frequencies between adjacent data with or without the same IDs in in-vehicle data sets, were found to be useful for improving the accuracy of attack detection, especially on detecting the DoS attack. For the usage of feature selection methods, the best performance model and important attributes towards different attacks in different data set were suggested, which could be used in future real world CAV attack detection.

Several limitations were found in this thesis. The first and most important limitation is the available data sets to cover possible CAV attacks. Because of the risk to conduct CAV cyber attacks in daily driving activities, all the data sets were from simulation or CAVs in a controlled environment. Moreover, the data sets only cover limited types of attacks in CAVs. As the technologies of CAVs are still evolving, attackers would conduct different types of attacks. More comprehensive data sets are needed to detect the attacks to CAVs more efficiently and comprehensively by overcoming these limitations. In addition, as in the thesis, only supervised machine learning models are used, more types of machine learning algorithms such as unsupervised algorithms could be used

on the detection. It could help to improve the detection rate of unseen attacks and also help to realize real time detection.

In this chapter, several future research directions were proposed and suggested in CAVCS. These research directions include: Collecting more comprehensive data sets so that more targeted attacks could be detected, and more 'real world' data could be used; Improving the performance of the models to conduct the detection more efficiently and precisely, which could also help to improve the detection rate of new attacks; Developing real time detection methods to help the CAVs detect the attack on time and respond to it appropriately; Developing other mitigation methods to enhance the framework of CAVCS, by which the CAVs could be safer and reliable.

Bibliography

[1] A. Nikitas, K. Michalakopoulou, E. T. Njoya, and D. Karampatzakis, "Artificial intelligence, transport and the smart city: Definitions and dimensions of a new mobility era," *Sustainability*, vol. 12, no. 7, p. 2789, 2020.

[2] H. Fan, F. Zhu, C. Liu, L. Zhang, L. Zhuang, D. Li, W. Zhu, J. Hu, H. Li, and Q. Kong, "Baidu apollo em motion planner," *arXiv preprint arXiv:1807.08048*, 2018.

[3] M. Dikmen and C. M. Burns, "Autonomous driving in the real world: Experiences with tesla autopilot and summon," in *Proceedings of the 8th International Conference on Automotive User Interfaces and Interactive Vehicular Applications*, ser. Automotive'UI 16. New York, NY, USA: ACM, 2016, pp. 225–228. [Online]. Available: http://doi.acm.org/10.1145/3003715.3005465

[4] Tesla, "Future of driving," Accessed Oct 18, 2018. [Online]. Available: http://www.tesla.com/model3

[5] L. Jones, "Driverless when and cars: where?[automotive autonomous vehicles]," *Engineering & Technology*, vol. 12, no. 2, pp. 36–40, 2017.

[6] B. J. Cottam, "Transportation planning for connected autonomous vehicles: How it all fits together," *Transportation Research Record*, p. 0361198118756632, 2018.

[7] B. Schoettle and M. Sivak, "A preliminary analysis of real-world crashes involving self-driving vehicles," *University of Michigan Transportation Research Institute*, 2015.

[8] K. Bimbraw, "Autonomous cars: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology," in *Informatics in Control, Automation and Robotics (ICINCO), 2015 12th International Conference on*, vol. 1. IEEE, 2015, pp. 191–198.

[9] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: opportunities, barriers and policy recommendations," *Transportation Research Part A: Policy and Practice*, vol. 77, pp. 167–181, 2015.

[10] X. Xu and C.-K. Fan, "Autonomous vehicles, risk perceptions and insurance demand: An individual survey in china," *Transportation Research Part A: Policy and Practice*, 2018.

[11] X. Kuang, F. Zhao, H. Hao, and Z. Liu, "Intelligent connected vehicles: the industrial practices and impacts on automotive value-chains in china," *Asia Pacific Business Review*, vol. 24, no. 1, pp. 1–21, 2018.

[12] J. Guanetti, Y. Kim, and F. Borrelli, "Control of connected and automated vehicles: State of the art and future challenges," *Annual Reviews in Control*, 2018.

[13] D. Li and H. Gao, "A hardware platform framework for an intelligent vehicle based on a driving brain," *Engineering*, vol. 4, no. 4, pp. 464–470, 2018.

[14] "Self Driving Vehicles in an Urban Context," 2015.

[15] GOV.UK. (2017, August) The key principles of vehicle cyber security for connected and automated vehicles. Accessed Dec 4, 2017. [Online]. Available: https://www.gov.uk/government/publications/principles-of-cyber-security-for-connected-and-automated-vehicles

[16] S. Levin and J. C. Wong, "Self-driving Uber kills Arizona woman in first fatal crash involving pedestrian," *The Guardian*, Mar. 2018. [Online]. Available: http://www.theguardian.com/technology/2018/mar/19/uber-self-driving-car-kills-woman-arizona-tempe

[17] "Tesla was on Autopilot in fatal crash," Mar. 2018. [Online]. Available: https://www.bbc.co.uk/news/world-us-canada-43604440

[18] "Autopilot Cited in Death of Chinese Tesla Driver - The New York Times," 2016. [Online]. Available: https://www.nytimes.com/2016/09/15/business/fatal-tesla-crash-in-china-involved-autopilot-government-tv-says.html

[19] "Hacker remotely crashes Jeep from 10 miles away - Telegraph," 2015. [Online]. Available: https://www.telegraph.co.uk/news/worldnews/northamerica/usa/11754089/Hacker-remotely-crashes-Jeep-from-10-miles-away.html

[20] T. K. S. Lab, "Tencent keen security lab: Experimental security research of tesla autopilot," 2019. [Online]. Available: https://keenlab.tencent.com/en/2019/03/29/Tencent-Keen-Security-Lab-Experimental-Security-Research-of-Tesla-Autopilot/

[21] J. Cui, L. S. Liew, G. Sabaliauskaite, and F. Zhou, "A review on safety failures, security attacks, and available countermeasures for autonomous vehicles," *Ad Hoc Networks*, 2018.

[22] H. M. Song, H. R. Kim, and H. K. Kim, "Intrusion detection system based on the analysis of time intervals of can messages for in-vehicle network," in *2016 international conference on information networking (ICOIN)*. IEEE, 2016, pp. 63–68.

[23] H. Zhou, W. Xu, Y. Bi, J. Chen, Q. Yu, and X. S. Shen, "Toward 5g spectrum sharing for immersive-experience-driven vehicular communications," *IEEE Wireless Communications*, vol. 24, no. 6, pp. 30–37, 2017.

[24] L. Liang, H. Ye, and G. Y. Li, "Toward intelligent vehicular networks: A machine learning framework," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 124–135, 2019.

[25] M. H. Eiza and Q. Ni, "Driving with sharks: Rethinking connected vehicles with vehicle cybersecurity," *IEEE Vehicular Technology Magazine*, vol. 12, no. 2, pp. 45–51, 2017.

[26] "Connected and autonomous vehicle cyber-security feasibility studies," 2019. [Online]. Available: https://apply-for-innovation-funding.service.gov.uk/competition/430/overview

[27] "Cyber security and space based services - ESA Business Applications," 2019. [Online]. Available: https://business.esa.int/funding/invitation-to-tender/cyber-security-and-space-based-services

[28] M. Mylrea, S. N. G. Gourisetti, and A. Nicholls, "An introduction to buildings cybersecurity framework," in *2017 IEEE symposium series on computational intelligence (SSCI)*. IEEE, 2017, pp. 1–7.

[29] M. Harb, A. Stathopoulos, Y. Shiftan, and J. L. Walker, "What do we (not) know about our future with automated vehicles?" *Transportation Research Part C: Emerging Technologies*, vol. 123, p. 102948, 2021.

[30] S. Chakraborty, M. A. Al Faruque, W. Chang, D. Goswami, M. Wolf, and Q. Zhu, "Automotive cyber–physical systems: A tutorial introduction," *IEEE Design & Test*, vol. 33, no. 4, pp. 92–108, 2016.

[31] N. Lyu, Z. Duan, L. Xie, and C. Wu, "Driving experience on the effectiveness of advanced driving assistant systems," in *2017 4th International Conference on Transportation Information and Safety (ICTIS)*. IEEE, 2017, pp. 987–992.

[32] M. Khan, "Truck platooning: Future of the freight industry," in *Transportation Association of Canada and ITS Canada 2019 Joint Conference and Exhibition*, 2019.

[33] E. E. Tsiropoulou, J. S. Baras, S. Papavassiliou, and S. Sinha, "Rfid-based smart parking management system," *Cyber-Physical Systems*, vol. 3, no. 1-4, pp. 22–41, 2017.

[34] L. F. P. de Oliveira, L. T. Manera, and P. D. G. da Luz, "Development of a smart traffic light control system with real-time monitoring," *IEEE Internet of Things Journal*, 2020.

[35] L. Yao, J. Wang, X. Wang, A. Chen, and Y. Wang, "V2x routing in a vanet based on the hidden markov model," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 889–899, 2017.

[36] SAE, *Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles*, jun 2018. [Online]. Available: https://doi.org/10.4271/J3016_201806

[37] J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, J. Z. Kolter, D. Langer, O. Pink, V. Pratt *et al.*, "Towards fully autonomous driving: Systems and algorithms," in *2011 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2011, pp. 163–168.

[38] Gov.UK, "Center for connected and autonomous vehicles," 2018. [Online]. Available: https://www.gov.uk/government/organisations/centre-for-connected-and-autonomous-vehicles

[39] "Connected and autonomous vehicle research and development projects," 2018. [Online]. Available: https://www.gov.uk/government/publications/connected-and-autonomous-vehicle-research-and-development-projects

[40] "Connected and autonomous Vehicles:The future ?" 2018. [Online]. Available: https://publications.parliament.uk/pa/ld201617/ldselect/ldsctech/115/115.pdf

[41] "Connected and autonomous vehicles bsi group," 2018. [Online]. Available: https://www.bsigroup.com/en-GB/Innovation/cav/

[42] "Connetced and autonomous vehicles catapult," 2018. [Online]. Available: https://ts.catapult.org.uk/innovation-centre/cav/

[43] *The Pathway to Driverless Cars Summary report and action plan*, Department for Transport, 2015.

[44] T. Luettel, M. Himmelsbach, and H.-J. Wuensche, "Autonomous ground vehicles—concepts and a path to the future," *Proceedings of the IEEE*, vol. 100, no. Special Centennial Issue, pp. 1831–1839, 2012.

[45] S. Narayanan, E. Chaniotakis, and C. Antoniou, "Factors affecting traffic flow efficiency implications of connected and autonomous vehicles: a review and policy recommendations," *Advances in Transport Policy and Planning*, vol. 5, pp. 1–50, 2020.

[46] D. J. Fagnant and K. M. Kockelman, "The travel and environmental implications of shared autonomous vehicles, using agent-based model scenarios," *Transportation Research Part C: Emerging Technologies*, vol. 40, pp. 1–13, 2014.

[47] D. Milakis, B. Van Arem, and B. Van Wee, "Policy and society related implications of automated driving: A review of literature and directions for future research," *Journal of Intelligent Transportation Systems*, vol. 21, no. 4, pp. 324–348, 2017.

[48] A. Taeihagh and H. S. M. Lim, "Governing autonomous vehicles: emerging responses for safety, liability, privacy, cybersecurity, and industry risks," *Transport reviews*, vol. 39, no. 1, pp. 103–128, 2019.

[49] N. Virdi, H. Grzybowska, S. T. Waller, and V. Dixit, "A safety assessment of mixed fleets with connected and autonomous vehicles using the surrogate safety assessment module," *Accident Analysis & Prevention*, vol. 131, pp. 95–111, 2019.

[50] R. Mariani, "An overview of autonomous vehicles safety," in *2018 IEEE International Reliability Physics Symposium (IRPS)*. IEEE, 2018, pp. 6A–1.

[51] N. H. T. S. Administration *et al.*, "Cybersecurity best practices for modern vehicles," *Report No. DOT HS*, vol. 812, no. 333, pp. 17–20, 2016.

[52] N. USDOT, "Automated driving systems: A vision for safety, september 2017," 2017.

[53] B. L. Bollinger, "The security and privacy in your car act: Will it actually protect you?" *North Carolina Journal of Law & Technology*, vol. 18, no. 5, p. 214, 2017.

[54] H. S. M. Lim and A. Taeihagh, "Autonomous vehicles for smart and sustainable cities: An in-depth exploration of privacy and cybersecurity implications," *Energies*, vol. 11, no. 5, p. 1062, 2018.

[55] ENISA, "Cyber security and resilience of smart cars." [Online]. Available: http://www.enisa.europa.eu/publications/cyber-security-and-resilience-of-smart-cars

[56] M. Goddard, "The eu general data protection regulation (gdpr): European regulation that has a global impact," *International Journal of Market Research*, vol. 59, no. 6, pp. 703–705, 2017.

[57] ENISA, "Good practices for security of smart cars." [Online]. Available: https://www.enisa.europa.eu/publications/smart-cars

[58] M. of Industry, I. T. of The People's Republic of China, and S. A. of the People's Republic of China, "Guideline for developing national internet of vehicles industry standard system." [Online]. Available: http://www.catarc.org.cn/upload/201802/13/201802131152200937.pdf

[59] K. Garidis, L. Ulbricht, A. Rossmann, and M. Schmäh, "Toward a user acceptance model of autonomous driving," in *Proceedings of the 53rd Hawaii international conference on system sciences*, 2020.

[60] J. Petit and S. E. Shladover, "Potential cyberattacks on automated vehicles." *IEEE Trans. Intelligent Transportation Systems*, vol. 16, no. 2, pp. 546–556, 2015.

[61] T. Perry, "Why the next denial-of-service attack could be against your car," *IEEE Spectrum: Technology, Engineering, and Science News*, 2016.

[62] M. Levi, Y. Allouche, and A. Kontorovich, "Advanced analytics for connected car cybersecurity," in *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*. IEEE, 2018, pp. 1–7.

[63] Q. He, X. Meng, and R. Qu, "Survey on cyber security of cav," in *Cooperative Positioning and Service (CPGP), 2017 Forum on.* IEEE, 2017, pp. 351–354.

[64] G. Sabaliauskaite and J. Cui, "Integrating autonomous vehicle safety and security," 11 2017.

[65] M. Islam, M. Chowdhury, H. Li, and H. Hu, "Cybersecurity attacks in vehicle-to-infrastructure (v2i) applications and their prevention," *CoRR*, vol. abs/1711.10651, 2017.

[66] S. Nie, L. Liu, and Y. Du, "Free-fall: Hacking tesla from wireless to can bus," *Briefing, Black Hat USA*, vol. 25, pp. 1–16, 2017.

[67] D. Satyajeet, A. Deshmukh, and S. Dorle, "Heterogeneous approaches for cluster based routing protocol in vehicular ad hoc network (vanet)," *International Journal of Computer Applications*, vol. 134, no. 12, pp. 1–8, 2016.

[68] M. N. Mejri, J. Ben-Othman, and M. Hamdi, "Survey on vanet security challenges and possible cryptographic solutions," *Vehicular Communications*, vol. 1, no. 2, pp. 53–66, 2014.

[69] M. S. Al-Kahtani, "Survey on security attacks in vehicular ad hoc networks (vanets)," in *Signal Processing and Communication Systems (ICSPCS), 2012 6th International Conference on.* IEEE, 2012, pp. 1–9.

[70] A. Shameli-Sendi, R. Aghababaei-Barzegar, and M. Cheriet, "Taxonomy of information security risk assessment (isra)," *Computers & security*, vol. 57, pp. 14–30, 2016.

[71] H.-K. Kong, T.-S. Kim, and M.-K. Hong, "A security risk assessment framework for smart car," in *2016 10th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS).* IEEE, 2016, pp. 102–108.

[72] M. Wolf and M. Scheibel, "A systematic approach to a qualified security risk analysis for vehicular it systems," *Automotive-Safety & Security 2012*, 2012.

[73] D. Ward, I. Ibarra, and A. Ruddle, "Threat analysis and risk assessment in automotive cyber security," *SAE International Journal of Passenger Cars-Electronic and Electrical Systems*, vol. 6, no. 2013-01-1415, pp. 507–513, 2013.

[74] T. Stolte, G. Bagschik, A. Reschka, and M. Maurer, "Hazard analysis and risk assessment for an automated unmanned protective vehicle," in *2017 IEEE Intelligent Vehicles Symposium (IV).* IEEE, 2017, pp. 1848–1855.

[75] T. Binion, J. Harr, B. Fields, S. Cielocha, and S. J. Balbach, "Risk assessment for an automated vehicle," Apr. 23 2015, uS Patent App. 14/057,467.

[76] C. L. Bates, S. P. Jones, E. J. Nelson, and J. M. Santosuosso, "Location-based vehicle risk assessment system," Mar. 11 2008, uS Patent 7,343,306.

[77] G. K. Rajbahadur, A. J. Malton, A. Walenstein, and A. E. Hassan, "A survey of anomaly detection for connected vehicle cybersecurity and safety," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 421–426.

[78] H. Lee, S. H. Jeong, and H. K. Kim, "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)*. IEEE, 2017, pp. 57–5709.

[79] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 3354–3361.

[80] J. Geyer, Y. Kassahun, M. Mahmudi, X. Ricou, R. Durgesh, A. S. Chung, L. Hauswald, V. H. Pham, M. Mühlegg, S. Dorn *et al.*, "A2d2: Audi autonomous driving dataset," *arXiv preprint arXiv:2004.06320*, 2020.

[81] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2446–2454.

[82] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang, "The apolloscape dataset for autonomous driving," *arXiv: 1803.06184*, 2018.

[83] Lyft, "Lyft level 5 open data." [Online]. Available: https://self-driving.lyft.com/level5/data/

[84] S. Agarwal, A. Vora, G. Pandey, W. Williams, H. Kourous, and J. McBride, "Ford multi-av seasonal dataset," *arXiv preprint arXiv:2003.07969*, 2020.

[85] F. Yu, H. Chen, X. Wang, W. Xian, Y. Chen, F. Liu, V. Madhavan, and T. Darrell, "Bdd100k: A diverse driving dataset for heterogeneous multitask learning," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2636–2645.

[86] G. Neuhold, T. Ollmann, S. Rota Bulo, and P. Kontschieder, "The mapillary vistas dataset for semantic understanding of street scenes," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4990–4999.

[87] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 3213–3223.

[88] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 621–11 631.

[89] G. J. Brostow, J. Fauqueur, and R. Cipolla, "Semantic object classes in video: A high-definition ground truth database," *Pattern Recognition Letters*, vol. 30, no. 2, pp. 88–97, 2009.

[90] A. Patil, S. Malla, H. Gang, and Y.-T. Chen, "The h3d dataset for full-surround 3d multi-object detection and tracking in crowded urban scenes," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 9552–9557.

[91] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 year, 1000 km: The oxford robotcar dataset," *The International Journal of Robotics Research*, vol. 36, no. 1, pp. 3–15, 2017.

[92] O. Yavanoglu and M. Aydos, "A review on cyber security datasets for machine learning algorithms," in *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017, pp. 2186–2193.

[93] A. Divekar, M. Parekh, V. Savla, R. Mishra, and M. Shirole, "Benchmarking datasets for anomaly-based network intrusion detection: Kdd cup 99 alternatives," in *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*. IEEE, 2018, pp. 1–8.

[94] T. Janarthanan and S. Zargari, "Feature selection in unsw-nb15 and kddcup'99 datasets," in *2017 IEEE 26th international symposium on industrial electronics (ISIE)*. IEEE, 2017, pp. 1881–1886.

[95] N. Moustafa and J. Slay, "Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set)," in *2015 military communications and information systems conference (MilCIS)*. IEEE, 2015, pp. 1–6.

[96] A. Patel, Q. Qassim, and C. Wills, "A survey of intrusion detection and prevention systems," *Information Management & Computer Security*, 2010.

[97] I. Butun, S. D. Morgera, and R. Sankar, "A survey of intrusion detection systems in wireless sensor networks," *IEEE communications surveys & tutorials*, vol. 16, no. 1, pp. 266–282, 2013.

[98] B. B. Zarpelão, R. S. Miani, C. T. Kawakani, and S. C. de Alvarenga, "A survey of intrusion detection in internet of things," *Journal of Network and Computer Applications*, vol. 84, pp. 25–37, 2017.

[99] A. Ferdowsi and W. Saad, "Generative adversarial networks for distributed intrusion detection in the internet of things," in *2019 IEEE*

Global Communications Conference (GLOBECOM). IEEE, 2019, pp. 1–6.

[100] J. Jabez and B. Muthukumar, "Intrusion detection system (ids): anomaly detection using outlier detection approach," *Procedia Computer Science*, vol. 48, pp. 338–346, 2015.

[101] F. Sabahi and A. Movaghar, "Intrusion detection: A survey," in *2008 Third International Conference on Systems and Networks Communications*. IEEE, 2008, pp. 23–26.

[102] S. Jose, D. Malathi, B. Reddy, and D. Jayaseeli, "A survey on anomaly based host intrusion detection system," in *Journal of Physics: Conference Series*, vol. 1000, no. 1. IOP Publishing, 2018, p. 012049.

[103] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," *Journal of Network and Computer Applications*, vol. 36, no. 1, pp. 16–24, 2013.

[104] H. Zhengbing, L. Zhitang, and W. Junqi, "A novel network intrusion detection system (nids) based on signatures search of data mining," in *First International Workshop on Knowledge Discovery and Data Mining (WKDD 2008)*. IEEE, 2008, pp. 10–16.

[105] S. Omar, A. Ngadi, and H. H. Jebur, "Machine learning techniques for anomaly detection: an overview," *International Journal of Computer Applications*, vol. 79, no. 2, 2013.

[106] R. Sekar, A. Gupta, J. Frullo, T. Shanbhag, A. Tiwari, H. Yang, and S. Zhou, "Specification-based anomaly detection: a new approach for detecting network intrusions," in *Proceedings of the 9th ACM conference on Computer and communications security*, 2002, pp. 265–274.

[107] W. Zhang, Q. Yang, and Y. Geng, "A survey of anomaly detection methods in networks," in *2009 International Symposium on Computer Network and Multimedia Technology*. IEEE, 2009, pp. 1–3.

[108] A. Shenfield, D. Day, and A. Ayesh, "Intelligent intrusion detection systems using artificial neural networks," *ICT Express*, vol. 4, no. 2, pp. 95–99, 2018.

[109] T. Lane and C. E. Brodley, "An application of machine learning to anomaly detection," in *Proceedings of the 20th National Information Systems Security Conference*, vol. 377. Baltimore, USA, 1997, pp. 366–380.

[110] S. Naseer, Y. Saleem, S. Khalid, M. K. Bashir, J. Han, M. M. Iqbal, and K. Han, "Enhanced network anomaly detection based on deep neural networks," *IEEE Access*, vol. 6, pp. 48 231–48 246, 2018.

[111] T. Shon and J. Moon, "A hybrid machine learning approach to network anomaly detection," *Information Sciences*, vol. 177, no. 18, pp. 3799–3821, 2007.

[112] A. Qayyum, M. Usama, J. Qadir, and A. Al-Fuqaha, "Securing connected autonomous vehicles: Challenges posed by adversarial machine learning and the way forward," *IEEE Communications Surveys Tutorials*, vol. 22, no. 2, pp. 998–1026, 2020.

[113] S. Kumar, K. Singh, S. Kumar, O. Kaiwartya, Y. Cao, and H. Zhou, "Delimitated anti jammer scheme for internet of vehicle: Machine learning based security approach," *IEEE Access*, vol. PP, pp. 1–1, 08 2019.

[114] G. Castignani, T. Derrmann, R. Frank, and T. Engel, "Smartphone-based adaptive driving maneuver detection: A large-scale evaluation study," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 9, pp. 2330–2339, 2017.

[115] M. Zhang, C. Chen, T. Wo, T. Xie, M. Z. A. Bhuiyan, and X. Lin, "Safedrive: online driving anomaly detection from large-scale vehicle data," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 2087–2096, 2017.

[116] N. T. Pham, E. Foo, S. Suriadi, H. Jeffrey, and H. F. M. Lahza, "Improving performance of intrusion detection system using ensemble methods and feature selection," in *Proceedings of the Australasian Computer Science Week Multiconference*, 2018, pp. 1–6.

[117] T. Bjerkestrand, D. Tsaptsinos, and E. Pfluegel, "An evaluation of feature selection and reduction algorithms for network ids data," in *2015 International Conference on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA)*, 2015, pp. 1–2.

[118] S. Aljawarneh, M. Aldwairi, and M. B. Yassein, "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model," *Journal of Computational Science*, vol. 25, pp. 152–160, 2018.

[119] S. Mohammadi, H. Mirvaziri, M. Ghazizadeh-Ahsaee, and H. Karimipour, "Cyber intrusion detection by combined feature selection algorithm," *Journal of information security and applications*, vol. 44, pp. 80–88, 2019.

[120] M. A. Ambusaidi, X. He, P. Nanda, and Z. Tan, "Building an intrusion detection system using a filter-based feature selection algorithm," *IEEE Transactions on Computers*, vol. 65, no. 10, pp. 2986–2998, 2016.

[121] G. WLG, "Connected and autonomous vehicles: A hacker's delight?" p. 20, 2017. [Online]. Available: https://gowlingwlg.com/GowlingWLG/media/UK/pdf/autodrive/170907-cyber-security-white-paper.pdf

[122] R. Von Solms and J. Van Niekerk, "From information security to cyber security," *computers & security*, vol. 38, pp. 97–102, 2013.

[123] D. Schatz, R. Bashroush, and J. Wall, "Towards a more representative definition of cyber security," *Journal of Digital Forensics, Security and Law*, vol. 12, no. 2, pp. 53–74, 2017.

[124] C. on National Security Systems, "National information assurance glossary," 2010. [Online]. Available: https://www.dni.gov/files/NCSC/documents/nittf/CNSSI-4009_National_Information_Assurance.pdf

[125] bsi group, "Pas 1885:2018 the fundamental principles of automotive cyber security. specification," 2018. [Online]. Available: https://shop.bsigroup.com/ProductDetail?pid=000000000030365446

[126] Gagandeep, Aashima, and P. Kumar, "Analysis of different security attacks in manets on protocol stack a-review," no. 5, pp. 269–275, 2012.

[127] K. Sahadevaiah and P. P. Reddy, "Impact of security attacks on a new security protocol for mobile ad hoc networks." *Network Protocols & Algorithms*, vol. 3, no. 4, pp. 122–140, 2011.

[128] G. Booch, *The unified modeling language user guide*. Pearson Education India, 2005.

[129] J. Ziegler, P. Bender, M. Schreiber, H. Lategahn, T. Strauss, C. Stiller, T. Dang, U. Franke, N. Appenrodt, C. G. Keller *et al.*, "Making bertha drive an autonomous journey on a historic route," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 2, pp. 8–20, 2014.

[130] S. Dolev, L. Krzywiecki, N. Panwar, and M. Segal, "Certificating vehicle public key with vehicle attributes a (periodical) licensing routine, against man-in-the-middle attacks and beyond," in *SAFECOMP 2013-Workshop ASCoMS (Architecting Safety in Collaborative Mobile Systems) of the 32nd International Conference on Computer Safety, Reliability and Security*, 2013.

[131] "The Newest Mobile Device: Self-driving Cars," 2018. [Online]. Available: https://www.chinalawinsight.com/2018/01/articles/corporate/the-newest-mobile-device-self-driving-cars/

[132] "One autonomous car will use 4,000 GB of data per day," 2018. [Online]. Available: https://www.networkworld.com/article/3147892/internet/one-autonomous-car-will-use-4000-gb-of-dataday.html

[133] "Addressing data privacy concerns in nhtsa v2v rules," 2017. [Online]. Available: https://us.eversheds-sutherland.com/NewsCommentary/Articles/197567/Addressing-Data-Privacy-Concerns-in-NHTSA-V2V-Rules

[134] S. Hansman and R. Hunt, "A taxonomy of network and computer attacks," *Computers & Security*, vol. 24, no. 1, pp. 31–43, 2005.

[135] M. Khurram, H. Kumar, A. Chandak, V. Sarwade, N. Arora, and T. Quach, "Enhancing connected car adoption: Security and over the air update framework," in *Internet of Things (WF-IoT), 2016 IEEE 3rd World Forum on*. IEEE, 2016, pp. 194–198.

[136] T. Ring, "Connected cars–the next targe tfor hackers," *Network Security*, vol. 2015, no. 11, pp. 11–16, 2015.

[137] "Why the Next Denial-of-Service Attack Could Be Against Your Car," 2016. [Online]. Available: https://spectrum.ieee.org/view-from-the-valley/transportation/safety/why-the-next-denial-of-service-attack-could-be-against-your-car

[138] R. N. Charette, "This car runs on code," *IEEE spectrum*, vol. 46, no. 3, p. 3, 2009.

[139] J. Moteff, "Risk management and critical infrastructure protection: Assessing, integrating, and managing threats, vulnerabilities and consequences." Library of Congress Washington DC Congressional Research Service, 2005.

[140] G. Stoneburner, A. Y. Goguen, and A. Feringa, "Sp 800-30. risk management guide for information technology systems," Gaithersburg, MD, United States, Tech. Rep., 2002.

[141] G. Tamasi and M. Demichela, "Risk assessment techniques for civil aviation security," *Reliability Engineering & System Safety*, vol. 96, no. 8, pp. 892 – 899, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0951832011000378

[142] W. F. Powers and P. R. Nicastri, "Automotive vehicle control challenges in the 21st century," *Control engineering practice*, vol. 8, no. 6, pp. 605–618, 2000.

[143] W. Stallings, *Cryptography and network security : principles and practice / Williams Stallings.*, 6th ed., Boston, Mass. ; London, 2014.

[144] C. Garling. (2017, October) Why cars will become the ultimate mobile device. Accessed Dec 4, 2018. [Online]. Available: https://builttoadapt.io/why-cars-will-become-the-ultimate-mobile-device-33dbaad40118

[145] H. C. Barbosa, D. A. Lima, A. M. Neto, G. B. Vitor, A. Martinesco, G. Rabelo, and V. H. Etgens, "The new generation of standard data recording device for intelligent vehicles," in *Intelligent Transportation Systems (ITSC), 2016 IEEE 19th International Conference on*. IEEE, 2016, pp. 2669–2674.

[146] P. Kohli and A. Chadha, "Enabling pedestrian safety using computer vision techniques: A case study of the 2018 uber inc. self-driving car crash," *arXiv preprint arXiv:1805.11815*, 2018.

[147] C. J. Jacobus and D. Haanpaa, "All weather autonomously driven vehicles," Jun. 5 2018, uS Patent 9,989,967.

[148] M. Konrad and M. Schramm, "Validation of adas by sensor fusion," *ATZ worldwide*, vol. 120, no. 6, pp. 56–59, 2018.

[149] "Waymo safety report: On the road to fully self-driving," 2017. [Online]. Available: https://www.auto-mat.ch/wAssets/docs/171019_waymo-safety-report-2017-10.pdf

[150] C. Wang, L. Yu, Y. Hao, and Wangc, "Automotive usability : Human computer interaction in the vehicle," 2012.

[151] J. Petit, B. Stottelaar, M. Feiri, and F. Kargl, "Remote attacks on automated vehicles sensors: Experiments on camera and lidar," *Black Hat Europe*, vol. 11, p. 2015, 2015.

[152] R. Kummerle, D. Hahnel, D. Dolgov, S. Thrun, and W. Burgard, "Autonomous driving in a multi-level parking structure," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 3395–3400.

[153] A. Ibisch, S. Stümper, H. Altinger, M. Neuhausen, M. Tschentscher, M. Schlipsing, J. Salinen, and A. Knoll, "Towards autonomous driving in a parking garage: Vehicle localization and tracking using environment-embedded lidar sensors," in *Intelligent Vehicles Symposium (IV), 2013 IEEE*. IEEE, 2013, pp. 829–834.

[154] L. Kong, M. K. Khan, F. Wu, G. Chen, and P. Zeng, "Millimeter-wave wireless communications for iot-cloud supported autonomous vehicles: Overview, design, and challenges," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 62–68, 2017.

[155] U. Farooq, M. Amar, E. ul Haq, M. U. Asad, and H. M. Atiq, "Microcontroller based neural network controlled low cost autonomous vehicle," in *Machine Learning and Computing (ICMLC), 2010 Second International Conference on*. IEEE, 2010, pp. 96–100.

[156] J. Choi, V. Va, N. Gonzalez-Prelcic, R. Daniels, C. R. Bhat, and R. W. Heath, "Millimeter-wave vehicular communication to support massive automotive sensing," *IEEE Communications Magazine*, vol. 54, no. 12, pp. 160–167, 2016.

[157] W.-J. Park, B.-S. Kim, D.-E. Seo, D.-S. Kim, and K.-H. Lee, "Parking space detection using ultrasonic sensor in parking assistance system," in *Intelligent Vehicles Symposium, 2008 IEEE*. IEEE, 2008, pp. 1039–1044.

[158] K. Zikidis, A. Skondras, and C. Tokas, "Low observable principles, stealth aircraft and anti-stealth technologies," *Journal of Computations & Modelling*, vol. 4, no. 1, pp. 129–165, 2014.

[159] J. Jackson, R. Saborio, S. A. Ghazanfar, D. Gebre-Egziabher, and B. Davis, "Evaluation of low-cost, centimeter-level accuracy oem gnss receivers," 2018.

[160] B. Hofmann-Wellenhof, H. Lichtenegger, and E. Wasle, *GNSS–global navigation satellite systems: GPS, GLONASS, Galileo, and more*. Springer Science & Business Media, 2007.

[161] M. L. Psiaki, T. E. Humphreys, and B. Stauffer, "Attackers can spoof navigation signals without our knowledge. here's how to fight back gps lies," *IEEE Spectrum*, vol. 53, no. 8, pp. 26–53, 2016.

[162] D. Margaria, E. Falletti, and T. Acarman, "The need for gnss position integrity and authentication in its: conceptual and practical limitations in urban contexts," in *Intelligent Vehicles Symposium Proceedings, 2014 IEEE*. IEEE, 2014, pp. 1384–1389.

[163] S. Zhang, J. Chen, F. Lyu, N. Cheng, W. Shi, and X. Shen, "Vehicular communication networks in the automated driving era," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 26–32, 2018.

[164] R. Molina-Masegosa and J. Gozalvez, "Lte-v for sidelink 5g v2x vehicular communications: a new 5g technology for short-range vehicle-to-everything communications," *IEEE Vehicular Technology Magazine*, vol. 12, no. 4, pp. 30–39, 2017.

[165] J. B. Kenney, "Dedicated short-range communications (dsrc) standards in the united states," *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.

[166] L. Chen and C. Englund, "Cooperative its—eu standards to accelerate cooperative mobility," in *Connected Vehicles and Expo (ICCVE), 2014 International Conference on*. IEEE, 2014, pp. 681–686.

[167] R. Zhengang and G. Yingbo, "Design of electronic toll collection system in expressway based on rfid," in *2009 International Conference on Environmental Science and Information Application Technology*. IEEE, 2009, pp. 779–782.

[168] L. Zhao, H.-S. Kang, and S.-R. Kim, "Improved clustering for intrusion detection by principal component analysis with effective noise reduction," in *Information and Communication Technology-EurAsia Conference*. Springer, 2013, pp. 490–495.

[169] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the kdd cup 99 data set," in *Computational Intelligence for Security and Defense Applications, 2009.* IEEE, 2009, pp. 1–6.

[170] H. Altwaijry and S. Algarny, "Bayesian based intrusion detection system," *Journal of King Saud University-Computer and Information Sciences*, vol. 24, no. 1, pp. 1–6, 2012.

[171] I. S. Arora, G. K. Bhatia, and A. P. Singh, "Comparative analysis of classification algorithms on kdd'99 data set," *International Journal of Computer Network and Information Security*, vol. 8, no. 9, p. 34, 2016.

[172] J.-H. Lee, J.-H. Lee, S.-G. Sohn, J.-H. Ryu, and T.-M. Chung, "Effective value of decision tree with kdd 99 intrusion detection datasets for intrusion detection system," in *Advanced Communication Technology, 2008. ICACT 2008. 10th International Conference on*, vol. 2. IEEE, 2008, pp. 1170–1175.

[173] "MIT Lincoln Laboratory: DARPA Intrusion Detection Evaluation," 1998. [Online]. Available: https://www.ll.mit.edu/ideval/docs/attackDB.html

[174] M. S. Al-Daweri, K. A. Zainol Ariffin, S. Abdullah *et al.*, "An analysis of the kdd99 and unsw-nb15 datasets for the intrusion detection system," *Symmetry*, vol. 12, no. 10, p. 1666, 2020.

[175] "Icsim simulator," 2020. [Online]. Available: https://github.com/zombieCraig/ICSim

[176] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[177] N. Bhargava, G. Sharma, R. Bhargava, and M. Mathuria, "Decision tree analysis on j48 algorithm for data mining," *Proceedings of International Journal of Advanced Research in Computer Science and Software Engineering*, vol. 3, no. 6, 2013.

[178] R. Sudrajat, I. Irianingsih, and D. Krisnawan, "Analysis of data mining classification by comparison of c4. 5 and id algorithms," in *IOP Conference Series: Materials Science and Engineering*, vol. 166, no. 1. IOP Publishing, 2017, p. 012031.

[179] I. H. Witten, E. Frank, M. A. Hall, and C. J. Pal, *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2016.

[180] T. R. Patil and S. Sherekar, "Performance analysis of naive bayes and j48 classification algorithm for data classification," *International Journal of Computer Science and Applications*, vol. 6, no. 2, pp. 256–261, 2013.

[181] F. Alam and S. Pachauri, "Detection using weka," *Advances in Computational Sciences and Technology*, vol. 10, no. 6, pp. 1731–1743, 2017.

[182] N. B. Amor, S. Benferhat, and Z. Elouedi, "Naive bayes vs decision trees in intrusion detection systems," in *Proceedings of the 2004 ACM symposium on Applied computing*. ACM, 2004, pp. 420–424.

[183] A. A. Salih and M. B. Abdulrazaq, "Combining best features selection using three classifiers in intrusion detection system," in *2019 International Conference on Advanced Science and Engineering (ICOASE)*, 2019, pp. 94–99.

[184] K. A. Taher, B. M. Y. Jisan, and M. M. Rahman, "Network intrusion detection using supervised machine learning technique with feature selection," in *2019 International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST)*. IEEE, 2019, pp. 643–646.

[185] W. Wang, X. Du, and N. Wang, "Building a cloud ids using an efficient feature selection method and svm," *IEEE Access*, vol. 7, pp. 1345–1354, 2018.

[186] M. A. Umar, C. Zhanfang, and Y. Liu, "A hybrid intrusion detection with decision tree for feature selection," *arXiv preprint arXiv:2009.13067*, 2020.

[187] J. Li and H. Liu, "Challenges of feature selection for big data analytics," *IEEE Intelligent Systems*, vol. 32, no. 2, pp. 9–15, 2017.

[188] A. Ali, S. Shaukat, M. Tayyab, M. A. Khan, J. S. Khan, J. Ahmad *et al.*, "Network intrusion detection leveraging machine learning and feature selection," in *2020 IEEE 17th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET)*. IEEE, 2020, pp. 49–53.

[189] M. Revathi and T. Ramesh, "Network intrusion detection system using reduced dimensionality," *Indian Journal of Computer Science and Engineering (IJCSE)*, vol. 2, no. 1, pp. 61–67, 2011.

[190] P. Chen, F. Li, and C. Wu, "Research on intrusion detection method based on pearson correlation coefficient feature selection algorithm," in *Journal of Physics: Conference Series*, vol. 1757, no. 1. IOP Publishing, 2021, p. 012054.