mathematics

Article

Machine Learning-Based Detection for Cyber Security Attacks on Connected and Autonomous Vehicles

Qiyi He [1,*], Xiaolin Meng [1], Rong Qu [2] and Ruijie Xi [1,3]

[1] Nottingham Geospatial Institute, University of Nottingham, Nottingham NG7 2TU, UK; xiaolin.meng@nottingham.ac.uk (X.M.); ruijie.xi1@nottingham.ac.uk (R.X.)

[2] School of Computer Science, University of Nottingham, Nottingham NG8 1BB, UK; rong.qu@nottingham.ac.uk

[3] GNSS Research Center, Wuhan University, Wuhan 430079, China

* Correspondence: qiyi.he@nottingham.ac.uk

Received: 2 June 2020; Accepted: 5 August 2020; Published: 7 August 2020

Abstract: Connected and Autonomous Vehicle (CAV)-related initiatives have become some of the fastest expanding in recent years, and have started to affect the daily lives of people. More and more companies and research organizations have announced their initiatives, and some have started CAV road trials. Governments around the world have also introduced policies to support and accelerate the deployments of CAVs. Along these, issues such as CAV cyber security have become predominant, forming an essential part of the complications of CAV deployment. There is, however, no universally agreed upon or recognized framework for CAV cyber security. In this paper, following the UK CAV cyber security principles, we propose a UML (Unified Modeling Language)-based CAV cyber security framework, and based on which we classify the potential vulnerabilities of CAV systems. With this framework, a new CAV communication cyber-attack data set (named CAV-KDD) is generated based on the widely tested benchmark data set KDD99. This data set focuses on the communication-based CAV cyber-attacks. Two classification models are developed, using two machine learning algorithms, namely Decision Tree and Naive Bayes, based on the CAV-KDD training data set. The accuracy, precision and runtime of these two models when identifying each type of communication-based attacks are compared and analysed. It is found that the Decision Tree model requires a shorter runtime, and is more appropriate for CAV communication attack detection.

Keywords: Connected and Autonomous Vehicle; cyber security; machine learning

1. Introduction

Connected and Autonomous Vehicles (CAVs) are a newly emerged research topic, which has rapidly attracted attention in both research and practice [1]. In the UK, the government set up a government centre called the "Centre for Connected and Autonomous Vehicles" in 2015 [2]. This centre published a report on Connected and autonomous vehicle research and development projects in 2018 [3]. The House of Lords also published the report "Connected and Autonomous Vehicles: The future" in 2017 [4]. Other organizations including the British Standard Institution (BSI) in the UK, also published a standards strategy report on CAVs in 2017 [5].

Some publications have also used the title of Connected and Automated Vehicles. For example, the Transport Systems Catapult [6], an innovation centre in the UK, used the term Automated at its website. As a rapidly developing subject, the naming of CAVs is not consistent in the literature, at present. In this paper, we 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 [7].

The Society of Automotive Engineers (SAE) has classified the automation of vehicles into six different levels, based on several criteria, including the capability to conduct simultaneously longitudinal or lateral driving tasks, the capability for objects and events detection and response, the capability of recovery when a system failure happens and the limitation of the operational design domain. At each different automation level, the duty of the driver and the CAV system differs. The details of level zero to level five automation are summarized in Table 1 [8].

Table 1. Details of Society of Automotive Engineers (SAE) automation levels (modified from [8]).

Level 0	No Driving Automation	Driver conducts all the vehicle motion control tasks. Driver is responsible for monitoring the surrounding objects and events and responding to them. If a system failure happens, the driver is responsible for recovering from it. There is no operational design domain at this automation level.
Level 1	Driver Assistance	The driver and system conduct the driving task together. The system is only capable of either longitudinal or lateral motion control. The driver is responsible for monitoring the surrounding objects and events. If a system failure happens, the driver is responsible for recovering from it. There is limited operational design domain at this automation level.
Level 2	Partial Driving Automation	The system is capable of simultaneous longitudinal and lateral motion control. The driver is responsible for monitoring the surrounding objects and events. If a system failure happens, the driver is responsible for recovering from it. There is limited operational design domain at this automation level.
Level 3	Conditional Driving Automation	The system is capable of simultaneous longitudinal and lateral motion control. The system is responsible for monitoring the surrounding objects and events and responding to them. If a system failure happens, the driver needs to be ready to respond to the system request or even take over driving the vehicles directly. There is limited operational design domain at this automation level.
Level 4	High Driving Automation	The system is capable of simultaneous longitudinal and lateral motion control. The system is responsible for monitoring the surrounding objects and events and responding to them. If a system failure happens, the system is responsible for recovering from it. There is limited operational design domain at this automation level.
Level 5	Full Driving Automation	The system is capable of simultaneous longitudinal and lateral motion control. The system is responsible for monitoring the surrounding objects and events and responding to them. If a system failure happens, the system is responsible for recovering from it. There is unlimited operational design domain at this automation level.

The distinct features of connectivity and autonomy, however, means that CAVs may be exposed to more cyber-attacks and, thus, are more vulnerable while exchanging data with their surrounding environment and other vehicles on the road [9]. In computer science, cyber security refers to the protection of a computer system's functions against cyber-attacks, including the damage to its hardware, software and data [10]. In CAVs, the cyber security concerns the protection of the CAV system against the cyber-attacks which compromise the CAV functions. Meanwhile, the cyber-attacks could be made both physically or remotely, in order to steal, alter or destroy the data in CAVs.

Likely being the biggest mobile device people will use in the near future, CAVs may cause severe consequences in people's lives, including not only private information leakage but also potential fatal physical damages. In early 2018, an Uber autonomous vehicle hit a cyclist during road testing [11].

It has also been reported that, in the USA [12] and China [13], Tesla vehicles have caused fatal incidents. Tesla announced that the driver's hands were not detected on the steering wheel for six seconds before the accident happened in the USA. Although it has been announced that the autopilot system was engaged, the vehicles should only be classified as a driver assistance system rather than a fully autonomous system, according to the definitions of automation levels in Table 1. In the USA, white-hat hackers have already attacked the Grand Cherokee successfully, taking control of the vehicle and manipulated its windows [14]. There is, therefore, a pressing demand to investigate CAV cyber security issues, even at early stages of development. In the UK, the CAV Standard Report has listed cyber security threats level to 'Very High' [15], and in August 2017, the UK government published the CAV Cyber Security Principles [16]. Due to the significant impact of CAVs on the daily lives of people, CAV cyber security should be considered with the highest priority and in a timely manner. These factors motivated the research conducted on the CAV cyber security framework in this paper.

In the developments of CAVs, developers are faced with complicated cyber security challenges. First, the characteristics of CAV cyber security mean that it is difficult to consider all the potential attacks before one happens. It is necessary for all the developers and users to be aware of that they have to constantly react to unknown attacks, as attack patterns are also evolving. Attackers only need to find one vulnerability gap to conduct an attack, while the defenders need to consider all the potential attacks to protect CAVs. Secondly, CAVs are built with a variety of components and functions. Even if only one of these breaks down, the whole system can fail. Vulnerability testing is thus complicated, given the many functions which work together in a complex CAV system. Thirdly, the different sensors in CAVs collect huge amounts of data, handling which is difficult, let alone the fact that the data are also collected in different forms. The format and content of the data should be compatible with CAV protocols, in order to facilitate easier data processing. Finally, CAVs communicate by various wireless communication technologies, such as Bluetooth, Dedicated Short-Range Communications (DSRC) and WiFi. Therefore, it is more difficult to prevent CAV cyber security threats, compared to those in wired networks.

In the existing literature, there have been discussions of the potential cyber security threats in CAVs and attempts to develop relevant frameworks to address them. However, there is still a lack of a widely adapted framework or model within which types and points of CAV cyber-attacks can be defined and classified consistently, as well as the prevention mechanisms being conducted effectively.

In this paper, we present a brief overview on the current status of CAV cyber security development, and build a Unified Modelling Language (UML)-based framework for CAV, following the UK CAV cyber security principles [16]. The new framework supports further analysis of potential cyber security threats in CAV systems. In addition, a new data set CAV-KDD is derived from the intrusion detection benchmark data set KDD99 [17]. In the new data set, the attacks which are not applicable to CAV and redundant in KDD99 are removed. The resulting CAV-KDD data set contains 14 communication-based sub-attacks in CAV. Two machine learning algorithms, namely Decision Tree and Naive Bayes, are tested on the new data set, and their accuracy, precision and runtime are compared. It is found that both algorithms have similar accuracy, while Decision Tree has a quicker runtime. However, both algorithms perform poorly when detecting unseen attacks. This presents an interesting topic for future work.

The rest of the paper is organized as follows: Section 2 presents a brief overview of related work on CAV cyber security. Section 3 defines the relationships between components in the CAV framework using UML, and explains each class in details. Potential attack points of CAVs are also defined, based on the new CAV cyber security UML framework. In Section 4, according to the new CAV framework, redundant types of cyber-attacks in the benchmark data set, KDD99, are removed. The processed new data set based on the CAV framework, named CAV-KDD, is then analysed statistically in Section 5. Two classification models are built using machine learning algorithms, and the performance of which is analysed, in terms of time, precision and accuracy, on detecting CAV cyber-attacks. Section 6 concludes the paper and provides the authors' recommendations by the authors for future work.

2. Related Work on CAV Cyber Security

As a newly emerged research topic, CAVs have recently caught an increasing amount of attention around the world. Governments, companies, research organizations, the media and the public have all paid great attention to the development of CAVs, and some progress has already been made.

In the USA, some states have already issued laws allowing CAV road tests [18]. Google [19] started testing Google driverless vehicles in 2009, established its subsidiary company Waymo in 2016 and has started its plan to allow a limited amount of people living in Phoenix to request driverless rides in 2018 (although there is still a safety supervising driver in the vehicle). Tesla [20] has also been developing CAV driving solutions on the road, and putting these technologies into commercial use. A large number of reports have been published at universities in the USA, such as the University of Michigan [21], which has a Mcity test field nearby. In Europe, traditional leading car manufacturers such as BMW, Audi and Mercedes Benz have all invested heavily in CAV development [22].

In China, the first CAV test field was built in Shanghai [23]. Baidu has launched their Apollo CAV platform, aiming to produce Level 3 autonomous vehicles by 2019 [24]. Traditional car manufacturers, including Changan, BYD, Guangzhou Automobile Group and Shanghai Automotive Industry Corp, have all announced their CAV development plans [25]. At a CAV competition held in China every year between universities [19], a communication platform is provided for the real use of CAVs, on which the flaws and advantages of CAV can be discovered, thus contributing to CAV research. IT companies including Alibaba [26] and Didi Chuxing [27] have also entered this competitive field.

In addition, public is witnessing research progress reported on websites and newspapers every day. According to the Boston Consulting Group Survey [28], people are willing to try and buy CAVs, and 55% of them said they would like to take a ride in a fully automated CAV. The majority of them would like to spend more than 5 thousand dollars for CAV functions in the vehicles.

In spite of the huge investments and resources on research and development of CAVs, relatively less focus has been placed on the security and privacy of CAV data. There are only a few works in the literature which are directly related to CAV cyber security.

Some initial attempts have been made to discuss the potential attacks on CAVs. In [29], potential CAV cyber-attacks are listed. It was concluded that GNSS spoofing and the injection of fake messages are among the most dangerous cyber threats. In [30], potential cyber-attacks were categorized into two main types; namely, passive attacks and active attacks. Passive attacks, such as eavesdropping and the release of information, are difficult to recognize but easy to defend against, as the attackers do not interact with the data; while active attacks, such as modification and spoofing, are easy to recognize but difficult to defend against, as attackers can modify or fake the messages in the data transmission. In [31], the authors pointed out that current vehicle safety standard ISO26262 does not consider the security issues to avoid both the unintentional and intentional attacks. Currently, there is no existing universal security or safety standard for CAVs. A systematic definition of attacks and attack analysis methods is, therefore, highly desirable for the development of CAVs.

In addition to the discussions of potential attacks on CAVs, other studies have discussed specific attacks on CAVs, aimed at proposing possible solutions using artificial intelligence. In [32], the authors comprehensively reviewed the current adversarial attacks on CAVs using machine learning algorithms. The potential attacks were also divided into the application layer, network layer, system level, privacy breaches, sensors attack and so on. In [32], the authors emphasised that the intrusion detection of cyber-attacks is of high importance in the development of CAVs.

In [33], 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 lower packet loss ratio. It was concluded that the machine learning-based scheme works effectively against the jamming attacks on the CAV location.

Based on the above literature, unlike cyber security in other fields such as mobile devices, CAV cyber-attacks could cause physical damages to users. According to a survey conducted at the

University of Michigan [34], the public is concerned more about physical damages caused by CAVs than the leakage of private information. However, it has been found that there is not enough related work on CAV cyber security. The European Space Agency (ESA) has recently opened a call for proposals for CAV cyber security solutions using artificial intelligence on CAV [35].

The identified gap in the current research could thus be summarized as follows: Firstly, there exists no systematic method to analyse the potential vulnerabilities of CAVs. Most of the literature has only focused on single specific attacks on CAVs, such as location spoofing attacks or adversarial attacks specific to algorithms in CAVs. It should also be noticed that there is also a lack of CAV cyber security data sets, as most research has focused on the theoretical aspects and there is a subsequent lack of detection methods.

To address this urgent research topic, in both industry and academia, a systematic method is needed to define the potential attacks and establish CAV cyber security data sets. In this paper, a UML-based CAV framework is built, in order to analyse the potential cyber security threats to CAVs, following the UK CAV cyber security framework to support the development of a systematic solution securing CAV systems and data transferred. A new data set, CAV-KDD, is derived for CAV cyber security detection. Two machine learning models are built based on Decision Tree and Naive Bayes, in order to compare their performance in detecting CAV cyber security attacks.

3. UML-Based CAV Cyber Security Framework

In June 2017, the UK government published an official document: "Key Principles of Vehicle Cyber Security for Connected and Automated Vehicles" [16]. In this document, the UK government published eight principles of CAV cyber security, covering the whole life cycle of CAVs, and providing protection guidance to sub-contractors, suppliers and potential third parties regarding hardware, software and data. We summarize and categorize these eight principles, the structure of which is presented in Figure 1.

As can be seen from Figure 1, Principle 1 is the most important, as it defines the requirements of top-level design concerning CAV cyber security. In addition to Principles 1.2 and 1.3, which consider human factors, Principle 1.4 (which considers security program design) is an essential step towards comprehensive protection. Principle 1.1, regarding the security program, divides the protection process into three stages:

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

2. When the attacks happen: The system should 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 should response to attacks appropriately and be able to recover from attacks.

In the current literature, there is no widely adapted framework for CAV cyber security [36], based on which attack points could be defined and efficient protection methods could be developed. According to the UK CAV cyber security principles we categorized in Figure 1, the most fundamental elements of CAV cyber security are the 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, the risks of the CAV system can be defined, assessed and managed (Principles 2.1 and 2.3). During the CAV operations, monitoring the CAV system can help to maintain security over its lifecycle (Principles 3.1 and 3.3). The CAV system could also respond to and support effective solutions appropriately after an attack (Principles 3.2 and 8).

The CAV cyber security can, thus, be divided into the security of hardware, software and data. In addition to hardware, software and data generated by CAVs, they are also connected to the outside world through 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.

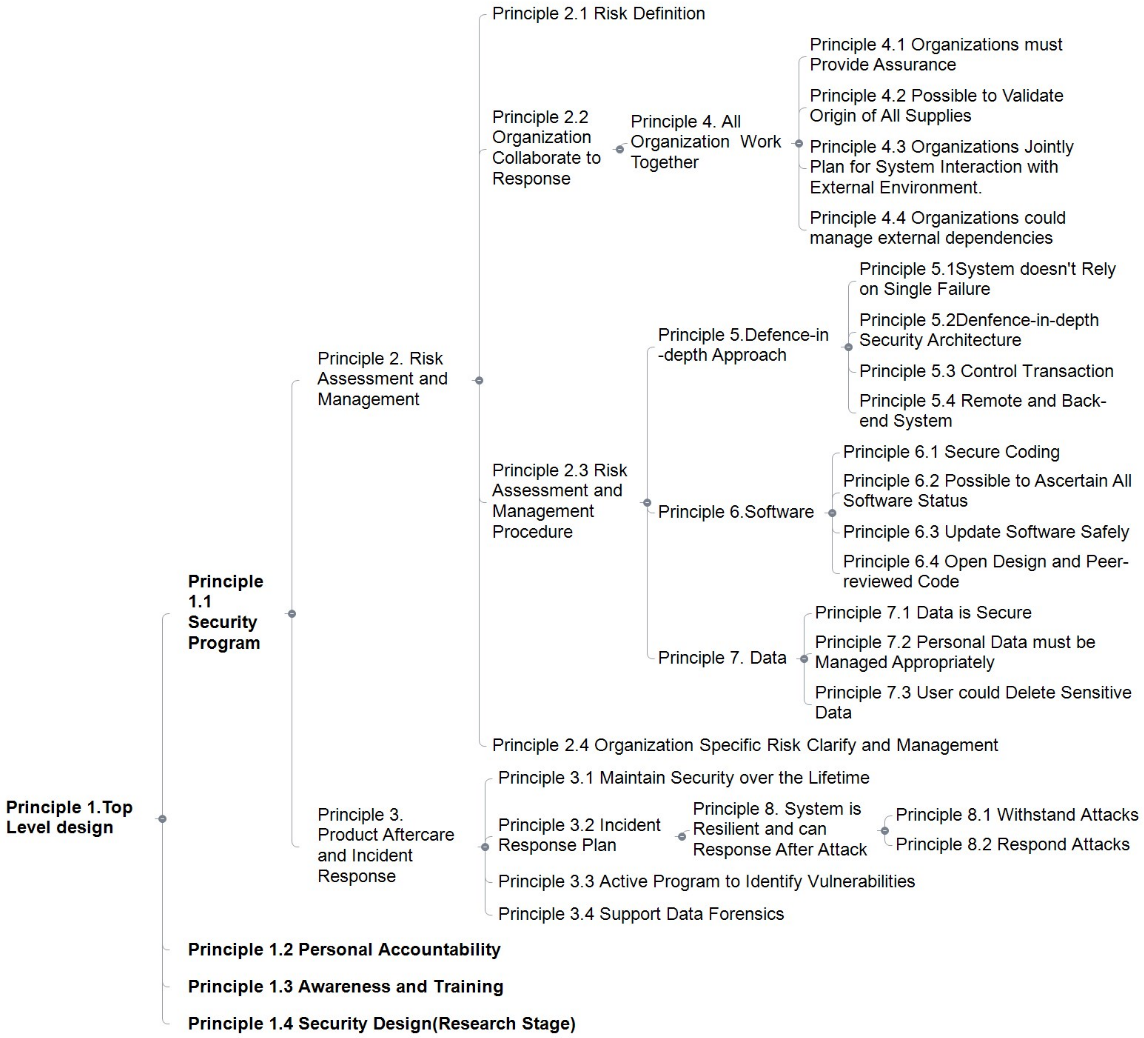
Principle 1.Top Level design
Principle 1.1 Security Program
Principle 2. Risk Assessment and Management
Principle 2.1 Risk Definition
Principle 2.2 Organization Collaborate to Response
Principle 4. All Organization Work Together
Principle 4.1 Organizations must Provide Assurance
Principle 4.2 Possible to Validate Origin of All Supplies
Principle 4.3 Organizations Jointly Plan for System Interaction with External Environment.
Principle 4.4 Organizations could manage external dependencies
Principle 2.3 Risk Assessment and Management Procedure
Principle 5.Defence-in -depth Approach
Principle 5.1System doesn't Rely on Single Failure
Principle 5.2Denfence-in-depth Security Architecture
Principle 5.3 Control Transaction
Principle 5.4 Remote and Back-end System
Principle 6.Software
Principle 6.1 Secure Coding
Principle 6.2 Possible to Ascertain All Software Status
Principle 6.3 Update Software Safely
Principle 6.4 Open Design and Peer-reviewed Code
Principle 7. Data
Principle 7.1 Data is Secure
Principle 7.2 Personal Data must be Managed Appropriately
Principle 7.3 User could Delete Sensitive Data
Principle 2.4 Organization Specific Risk Clarify and Management
Principle 3. Product Aftercare and Incident Response
Principle 3.1 Maintain Security over the Lifetime
Principle 3.2 Incident Response Plan
Principle 8. System is Resilient and can Response After Attack
Principle 8.1 Withstand Attacks
Principle 8.2 Respond Attacks
Principle 3.3 Active Program to Identify Vulnerabilities
Principle 3.4 Support Data Forensics
Principle 1.2 Personal Accountability
Principle 1.3 Awareness and Training
Principle 1.4 Security Design(Research Stage)

Figure 1. Structure of UK Connected and Autonomous Vehicle (CAV) cyber security principles (structured from [16]).

Unified Modelling Language (UML) has been widely used in software engineering to define and model the structures of systems [37]. In UML, a class diagram is used to build the concept structure of a system, showing both the main components of the system and their relationships with other components.

As shown in Figure 2, the proposed UML-based CAV cyber security framework is developed to define the relationships between each component and the structure in the CAV, including hardware, software and their generated data, to help the vehicle to function well. Based on the framework, different types and points of potential CAV cyber-attacks can be analysed and categorized. The main classes in this UML-based CAV framework include Vehicle Data, Data Processor and Vehicle Functions.

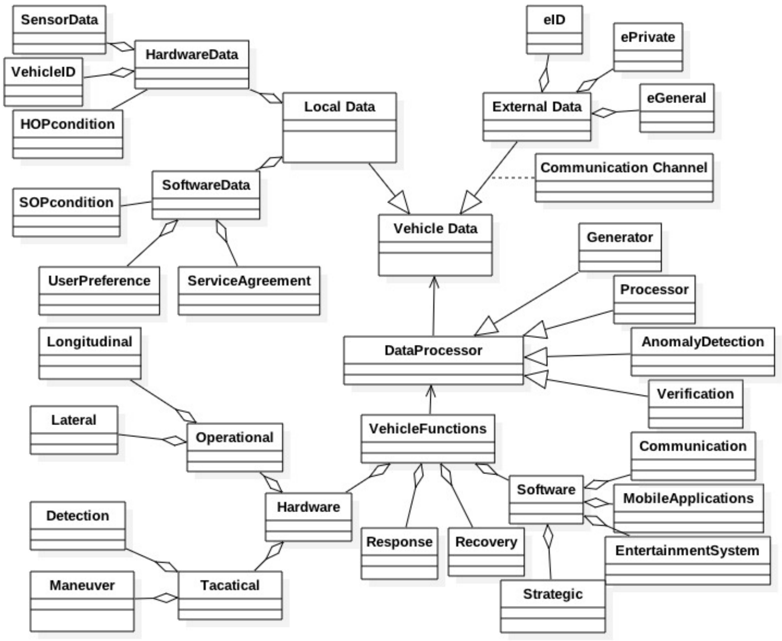

Figure 2. Unified Modelling Language (UML)-based CAV framework.

3.1. Vehicle Data

CAVs make decisions and implement relevant vehicle functions based on data. Thus, Vehicle Data is the most fundamental component in the CAV framework. In the Vehicle Data class, the data can be divided into local data and external data. The Vehicle Data class refers to Principles 5 to 7 in Figure 1.

Local Data has two sub-classes, which are hardware data and software data in the CAV framework. These two sub-classes include not only data generated by hardware and software, but also the operating condition data of hardware and software. The HardwareData class is the sensor data collected from the vehicle's surroundings by various CAV sensors, including Radar, Global Navigation Satellite Systems (GNSS) and camera [38]; for example, GNSS and image data which determine the current position of a CAV. In addition, the VehicleID class contains data identifying the vehicle, such as the electric plate (a unique number or letters assigned by government department). As CAVs exchange data and information with other entities, including other CAVs, infrastructures and pedestrians, VehicleID also contains a unique pair, the public key and private key, which are used to encrypt messages and check identification of vehicles [39]. The HOPcondition class is the operation condition data of hardware.

The SoftwareData class is comprised of Local Data collected by the software in CAVs, such as the onboard entertainment system. CAVs will very likely be an important smart mobile device people use in the future [40]. 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'. The UserPreference class contains such preference data of users, based on which CAVs make the best decision for the specific users. The ServiceAgreement class defines protocols that the software will comply to, including privacy protection and the protocols of other services. The SOPcondition class provides the operating condition data of the software.

The External Data class is comprised of data received from other entities, such as other CAVs and intelligence infrastructures in the communication network. All data are received through communication channels such as Vehicle to Vehicle (V2V), Vehicle to Infrastructure (V2I) communication, which comprise the Communication Channel class. As each entity has its own ID stored in its local data, the external data also needs this information to guarantee the identification of data sender, while the eID class contains ID information of the sender. In the external data, after identifying the sender ID, messages are 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 user preferences. This can only be accessed by specific users and is stored in the ePrivate class. The eGeneral class stores data that everyone could access, such as position and vehicle size data.

3.2. Data Processor

CAVs deal with massive amounts of data every day. It has been reported that each CAV produces up to 4000 GB during just one hour of driving [41]. In addition, adding a V2V communication system to a vehicle may require 10 messages per second [42], which also increases the workload of the data processors. How the data are processed is even more important than how they are collected. CAVs are equipped with a data processor to clean data and support making appropriate decisions. The Data Processor class is related to Principles 2.3, 3.1 and 3.3 in Figure 1.

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

3.3. Vehicle Functions

If there is no anomalous behaviour in the CAV system, after being processed, relevant data will be used to make decisions using the Vehicle Functions class. The Vehicle Functions class is related to Principles 3.1, 3.2, 5, 6 and 8 shown in Figure 1, and are defined accordingly as shown in Figure 2.

The functions of CAVs can be divided into Hardware and Software classes in the CAV framework, as shown in Figure 2. A variety of different driving tasks and operations have been categorized based on SAE J3016.

In SAE J3016 [8], the dynamic driving tasks of a vehicle are divided into three types, namely operational functions, tactical functions and strategic functions, where the former two belong to the Hardware class and the latter to the Software class. Operational functions comprise the basic vehicle motions, such as the longitudinal and lateral movements. Tactical functions perform the monitoring of surrounding environments and the associated responses, as well as manoeuvre planning. Operational functions and tactical functions may have some overlaps. Strategic operations involve route planning. Currently, the strategic operations are not included in the dynamic driving tasks in J3016 categories.

After a CAV detects its surrounding objects, it uses operational functions to respond. Based on SAE J3016, the Hardware class can be divided into Operational and Tactical classes. The Operational class has two sub-classes, which are longitudinal and lateral. These two sub-classes include relevant hardware functions when the vehicle undergoes longitudinal or lateral motions. The Tactical class also has two sub-classes: The Detection class is for the monitoring of surrounding objects and events through sensors including Radar, LiDAR and cameras. The Manoeuvre class is to take relevant manoeuvres such as turning the indicators on.

In addition to hardware functions, software functions such as entertainment system and mobile applications functions are also an essential part of CAVs. Based on this, the Software class contains the entertainment system and mobile applications. In addition, the Communication class supports all of the data receiving and sending functions. The Strategic class plans 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 the Hardware and Software classes, the Response class takes relevant actions based on the data from the hardware and software. The Recovery class is for fallback when a system failure happens, making sure CAVs are resilient and fail-safe.

3.4. Possible Attack Points

Cyber-attacks in computer networks can be categorized into different types including viruses, worms, buffer overflows, DoS attacks, network attacks, physical attacks, password attacks and information gathering attacks [43]. In traditional automobile vehicles, the points of attacks have been categorized into two types [44], namely attacks to the audio system or mobile applications, and attacks to the Controller Area Network (CAN), which is an inner vehicle communication network for micro-controllers and devices. As the CAN is connected to all of the in-vehicle hardware components including the brakes, air conditioning, steering 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 are also connected within the overall transportation infrastructure; thus, all of the above attacks to automobiles could happen in CAV. Moreover, with the increasing amount of autonomy and connectivity functions, there will be more vulnerabilities or attack points. Cyber security in CAV is necessary to protect the CAV system from cyber-attacks affecting their performance, either remotely or physically. It is necessary to identify, define and classify possible types of attacks to CAVs at an early stage. Based on the UML-based CAV framework established in Figure 2, 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 brakes. It has already been reported that hackers could take control of the brakes or air conditioners of Nissan [45] and JEEP vehicles. JEEP even recalled more than 1.4 million vehicles to install security patches, due to this type of hacking [46]. 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 on activities.

There are several attack points on the CAVs hardware. The mainstream sensors on CAVs include cameras, Light Detection and Ranging (LiDAR) and radars as included in Table 2. All of these sensors could be attacked either physically or remotely; for example, the cameras could be misled by fake images or the radar signal could be jammed. Attackers could even hack into the camera system, in order to monitor the vehicle's activities. Moreover, the GNSS system could also be attacked by experienced attackers. For example, the GNSS system could be jammed, following which the vehicle may not receive a GNSS signal for navigation or locating its position.

2. Vehicle software. CAVs could be installed with more than 100 million lines of code, while Boeings new 787 dreamliners are equipped with only 6.5 million lines of code [47]. This leads to a higher number of vulnerabilities in CAVs. The entertainment system, the installed mobile applications and the audio system onboard could all provide potential attack points for attackers. After taking control of the software, the data exchange could be monitored or the hardware could even be harmed if software is taken control of.

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

4. Communication channel. Potential attacks may also target at the communication channels. The attack points can be through Vehicle to Vehicle (V2V), Vehicle to Infrastructure (V2I), Vehicle to Cloud (V2C) or Vehicle to Everything (V2X) communication. A communication channel can easily be blocked, if attackers send a huge amount of messages at the same time. In addition, eavesdropping in communication channels can also cause information leakage.

Based on these analyses, the possible attack points to CAVs are summarized in Table 2. As the technologies adapted to 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 framework is extendable to include and categorise different types of new attacks.

Table 2. Possible attack points.

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

In this paper, the research focuses are on structuring the UK Cyber Security Principles [16], in order to build a CAV cyber security framework categorising communication-based attacks made remotely through communication channels to CAVs. Machine learning techniques are then demonstrated to classify these cyber-attacks. The points of attacks are shown in Figure 3, where CAVs exchange data with their surrounding environments using V2V, V2I and V2C communication channels.

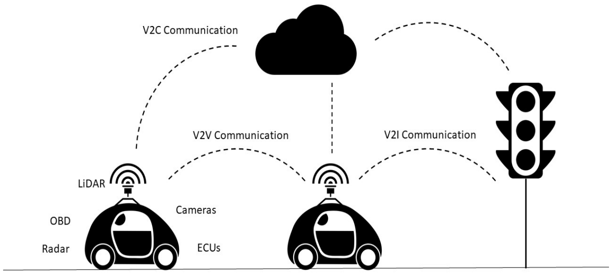

Figure 3. Attack points through communications.

4. The New CAV Cyber-Attack Data Set CAV-KDD

As a rapidly developing topic, CAVs are yet to be fully developed before they can drive safely on the road. In the existing literature on CAVs, it is difficult to access and obtain well processed and labelled data sets, especially relating to CAV cyber-attacks. In this paper, we adapt the widely used KDD99 benchmark data set [48] on network intrusion detection, and 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 Section 3.

4.1. 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 [48]. 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 [49].

KDD99 has approximately 5 million data records, each with 42 attributes. The 42nd attribute is the label of either normal or attack. KDD99 also provides a 10% data set with about 500 thousands data records for training and testing, for those who find the original data set too big for data processing. The attacks in KDD99 are comprised of four major types with 39 sub-attacks [50], as follows [51]:

1. PROBE, which is the Probing attacks. This type of attack monitors or scans the system for 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 of the resources, such that the system or communication channel is 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 attack. Attackers conducting U2R attacks aim to gain access to superuser accounts. They discover 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 a remote connection. The attacker does not have an authorized account in the system, but can gain local access to it. 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 we propose and test in Section 5.

KDD99 provides a comprehensive data set that covering a variety of attack types in computer networks. However, the data set cannot be used directly for CAV cyber security, due to the distinct characteristics of CAVs mentioned above. In this paper, we adapt and process the KDD99 data set by removing irrelevant attack types, based on the CAV framework established and possible attack points identified in Sections 3. The possible attack types in KDD99 which may also happen in CAV are shown in Table 3.

In Table 3, 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 was reduced from 39 to 14, with 19 types of possible CAV attack and 6 types of irrelevant attack. The justifications of data processing on the attack types are listed as follows.

1. Some attacks were without a clear definition. As the data are from the KDD99 data set, the definitions of attacks refer 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 [52]. All the descriptions of the attacks are referenced from the official description at the MIT Lincoln Lab web site [53]. Some sub-attacks lacked clear definitions and, thus, could not be classified as type P in CAV cyber-attacks. Their attack type could be changed, once a clear definition is available.

2. Some attacks do not fit into the CAV cyber security framework. In Section 3, a UML-based CAV framework is built to define different data in CAV communication and functions. However, as KDD99 is a data set on computer and network security, their 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. These types of attacks do not fit into the CAV framework and, thus, were removed.

3. Some attacks were not compatible with the CAV attack points. To conduct an attack, except for the physical damage, attackers need to find one of the vulnerable points (as identified in Section 3) in a

CAV system. These attack points could be in physical parts, software, data or communication channel. In KDD99, some attacks can only happen under specific conditions and platforms which, thus, are not applicable to the CAV attack points. The possibilities of these attacks in CAV are low; for example, the apache2 attack can only happens in an Apache Web Server. If a CAV does not use the Apache Web Server, the attack cannot be conducted.

Table 3. KDD99 sub-attacks possible in CAVs.

	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			

5. Experiments

In the CAV framework built in Section 3, anomaly detection is an important part. Two machine learning algorithms were developed in Weka [54] to build two classification models, Naive Bayes and Decision Tree, to detect anomalous behaviours in the data. The experiments were 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 University of Waikato, which has been widely used in industry and research to conduct analysis and develop machine learning models.

5.1. CAV-KDD Data Preprocessing

The KDD99 data set has more than 4 million data records, and is too big for data processing on personal computers. In this paper, the training data set with 10% of the KDD99 data set was used. After removing duplicates and irrelevant attack types, a new data set, which was compatible with the new CAV cyber security framework, named CAV-KDD, was established. The amount of normal data and attack data in both the training and testing data sets is presented in Tables 4 and 5.

Table 4. Amount of normal and attack data in the training data sets.

	10% KDD99 Data	CAV-KDD Data
Attacks	396,743	54,485
Normal	97,278	87,832
Total	494,021	142,317

Table 5. Amount of normal and attack data in the testing data sets.

	10% KDD99 Test Data	CAV-KDD Test Data
Attacks	250,436	23,348
Normal	60,593	47,913
Total	311,029	71,261

In addition, to amount of each sub-attack type in the CAV-KDD training and testing sets are shown in Table 6.

Table 6. Amount of sub-attacks types 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

The CAV-KDD data was then preprocessed in Weka, in the following steps:

1. The normal and 14 sub-attacks were labeled as 0 to 14, as shown in Table 3.

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

3. The data then needed to be discretized. The unsupervised-attribute-discretize algorithm in Weka was used to discretize the normalized data. For other categorized attribute data such as protocol_type or service, the unsupervised-attribute-numerictonominal algorithm was used.

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

5.2. Experiment Methods

In Weka, the machine learning algorithms Naive Bayes and J48 were used to build the two classification models Naive Bayes and Decision Tree to classify and detect CAV cyber-attacks.

Decision Tree is one of the most-used classification models, with good readability [55]. 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 with respect to an attribute, while 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 classification.

In Weka, the J48 algorithm uses the C4.5 technique to build the decision tree. 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 (1), as follows [56]:

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

where n is the number of partitions (classification labels) of the data set and p_i refers to the proportion of the ith partition. Thus, the information gain can be calculated by Equation (2), as follows:

$$Gain(V,a) = Entropy(V) - \sum_{j=1}^{J} \frac{|V_j|}{|V|} Entropy(V_j), \tag{2}$$

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 can be calculated by Equation (3), as follows:

$$GainRatio(V,a) = \frac{gain(V,a)}{IV(a))}, \tag{3}$$

in which, the intrinsic value (IV) is calculated in Equation (4) as follows:

$$IV(a) = -\sum_{j=1}^{J} \frac{|V_j|}{|V|} log_2 \frac{|V_j|}{|V|}. \tag{4}$$

Then, each value of the attribute becomes a branch of this tree and the data are split into different classes or tree leaves. The process will be repeated until the information gain ratio reaches the threshold [57], which is set to 0.25 as default in this experiment. In the 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, with the highest information gain, was chosen to be the root node.

Naive Bayes was built based on the Bayesian probability model. It assumes that all the attributes in the data are independent, meaning that each attribute has no impact on the other attributes [58]. The Naive Bayes model calculates the conditional probabilities of classes, the class with a highest probability being the prediction result [59]. The equation of Naive Bayes is presented in Equation (5) as follows [60]:

$$P(c|X) = \frac{P(X|c)P(c)}{P(X)}, \tag{5}$$

where $P(c|X)$ is the posterior probability of class c under the 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. In CAV-KDD, c is the label of normal or attack data and X is the data set of 39 chosen attributes. Based on attributes of each data point in the testing data set, the probabilities of it belonging to different labels are calculated. Each data then is classified to the label with the highest probability.

5.3. Experiment Results

As mentioned in Section 4, after processing the original KDD99 data, the number of attack types was reduced to 14 in CAV-KDD. We used CAV-KDD to build the detection models, which were tested on the CAV-KDD testing data set. To avoid the overfitting problem, the training set firstly uses 10-folds validation to build the model. Then the machine learning model is validated in the CAV-KDD testing data set. The overall accuracy, precision and runtime of the Decision Tree and Naive Bayes net models are compared in Table 7. In this paper, the accuracy indicates the ratio of correct classified attacks in the total number of classification.

Table 7. Accuracy and runtime of J48 and Naive Bayes.

	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
J48	99.80%	97.04%	2.42	0.94

From Table 7, 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 false positive (FP) rate of attack classification is also a crucial metric to evaluate the performance of the models. In real-world situations, if a machine learning model classifies the attack data as normal data, the consequences could be life-threatening. Based on this, the false positive rate is shown in Table 8. The precision of each model based on the following Equation (6) could also be analysed, as shown in Table 8.

$$Precision = \frac{TP}{TP + FP} \tag{6}$$

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 are 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.

Table 8. False positive rate of J48 and Naive Bayes.

	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%
J48	0.1%	5.6%	94.64%

The accuracy and false positive rate of detecting normal and anomalous data of each sub-type of attacks are listed in Table 9.

Table 9. Accuracy of sub-attack types obtained by two models.

			J48 Accuracy	J48 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 9, it can be seen that both machine learning classification models had high accuracy when identifying CAV cyber-attacks. The false positive 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 is noticeable that both machine learning algorithms performed poorly on attack types which were only included in the testing data set; namely mailbomb, 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 our future research.

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

6. Summary and Future Work

CAV technologies are becoming more advanced and mature now. It is believed that CAVs will be on the road for commercial uses as early as 2025. However, issues in CAV cyber security have not been considered as much as other CAV technologies, thus being of increasingly critical importance and high priority in current CAV developments. Cyber-attacks in CAVs may cause serious consequences, not only relating to the leakage of personal information but also to physical injuries or even fatalities. The importance of CAV cyber security has been highly emphasized by the UK organizations and the government.

In this paper, we analysed different types of CAV communication-based cyber security attacks, and established a UML-based CAV framework with different components, based on the UK CAV Cyber Security Principles. Using this CAV framework as guidance, possible CAV attack points were assessed and categorized.

A new data set, named CAV-KDD, was built based on the 10% KDD99 benchmark data set. Among the 39 types of cyber-attacks in the original KDD99 data set, the irrelevant attacks and undefined attacks were removed based on the proposed CAV cyber security framework, leading to 14 types of CAV cyber-attacks in the CAV-KDD data set. A large amount of redundant normal and attack data were also removed from the original KDD99 data set.

The newly established CAV-KDD data set was then statistically analysed using two machine learning algorithms, namely the Naive Bayes and Decision Tree, to test the accuracy of CAV cyber-attack detection by the two classification models. Naive Bayes was more accurate than Decision Tree when identifying the PROBE attacks, while Decision Tree obtained a higher accuracy when identifying the DoS attacks. Both models performed poorly when detecting U2R and R2L attacks. However, both algorithms had similar accuracy in detecting the 14 attacks, and Decision Tree had a shorter runtime. Based on the results, Decision Tree was shown to be more appropriate for detecting CAV communication-based attacks.

It was found that the classification models did not perform well on new types of unseen CAV cyber-attacks; that is, those which were not included in the training data set. Furthermore, both models performed poorly when detecting the U2R and R2L types of attacks. A high accuracy of detection is crucially important for CAVs to be on roads safely. In our future work, feature selection methods and hybrid methods will be used to further improve the accuracy and reduce the runtime. The combination of supervised and unsupervised machine learning algorithms will also be investigated, in order to improve the accuracy of identifying unseen attacks. The performance metric of classification models could also be improved, regarding different types of data. In addition, the attacks discussed in this paper only included communication-based attacks and not physical attacks. The CAV-KDD data set, therefore, does not contain all of the possible recognized attack types to CAV. The different types of data are imbalanced in the data set as well. Additionally, the technologies of CAVs are still evolving. The computing capability can be increased when more advanced computing units are adapted to CAVs. The detection and assessment of physical cyber-attacks and new types of attacks, as well as the improvements of detection machine learning models, present other interesting research topics in future work regarding CAVs.

Author Contributions: Conceptualization, Q.H.; Formal analysis, Q.H.; Methodology, Q.H., X.M., R.Q. and R.X.; Supervision, X.M. and R.Q.; Writing–original draft, Q.H.; Writing–review and editing, Q.H., X.M., R.Q. and R.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Guerra, E. Planning for cars that drive themselves: Metropolitan Planning Organizations, regional transportation plans, and autonomous vehicles. *J. Plan. Educ. Res.* **2016**, *36*, 210–224.
2. Gov.UK, Center for Connected and Autonomous Vehicles, 2018. Available online: https://www.gov.uk/government/organisations/centre-for-connected-and-autonomous-vehicles (accessed on 9 December 2018).
3. Connected and Autonomous Vehicle Research and Development Projects, 2018. Available online: https://www.gov.uk/government/publications/connected-and-autonomous-vehicle-research-and-development-projects (accessed on 11 December 2018).

4. Connected and Autonomous Vehicles: The Future? 2018. Available online: https://publications.parliament.uk/pa/ld201617/ldselect/ldsctech/115/115.pdf (accessed on 12 December 2018).
5. Connected and Autonomous Vehicles Bsi Group, 2018. Available online: https://www.bsigroup.com/en-GB/Innovation/cav/ (accessed on 12 December 2018).
6. Connetced and Autonomous Vehicles Catapult, 2018. Available online: https://ts.catapult.org.uk/innovation-centre/cav/ (accessed on 16 December 2018).
7. The Pathway to Driverless Cars Summary Report and Action Plan, 2018. Available online: https://assets.publishing.service.gov.uk/government/uploads/system/uploads/attachment_data/file/401562/pathway-driverless-cars-summary.pdf (accessed on 6 August 2020).
8. Taxonomy and Definitions for Terms Related to On-Road Motor Vehicle Automated Driving Systems, 2018. Available online: https://www.sae.org/standards/content/j3016_201806/ (accessed on 2 January 2019).
9. Parkinson, S.; Ward, P.; Wilson, K.; Miller, J. Cyber threats facing autonomous and connected vehicles: Future challenges. *IEEE Trans. Intell. Transp. Syst.* **2017**, *18*, 2898–2915.
10. Schatz, D.; Bashroush, R.; Wall, J. Towards a more representative definition of cyber security. *J. Digit. Forensics Secur. Law.* **2017**, *12*, 8.
11. Levin, S.; Wong, J.C. Self-driving Uber kills Arizona woman in first fatal crash involving pedestrian. *The Guardian*, 19 March 2018.
12. Tesla was on Autopilot in Fatal Crash, 2018. Available online: https://www.bbc.co.uk/news/world-us-canada-43604440 (accessed on 15 June 2018).
13. Autopilot Cited in Death of Chinese Tesla Driver. *The New York Times*, 14 September 2016.
14. Hacker remotely crashes Jeep from 10 miles away. *Telegraph*, 21 July 2015.
15. CAV Standards Strategy Summary Report, 2017. Available online: https://www.bbc.co.uk/news/world-us-canada-43604440 (accessed on 15 June 2018).
16. GOV.UK. *The Key Principles of Vehicle Cyber Security for Connected and Automated Vehicles*; 2017. Available online: https://www.gov.uk/government/publications/principles-of-cyber-security-for-connected-and-automated-vehicles/the-key-principles-of-vehicle-cyber-security-for-connected-and-automated-vehicles (accessed on 6 August 2020).
17. UCI kdd cup 1999 Data Data Set, 1999. Available online: https://archive.ics.uci.edu/ml/datasets (accessed on 1 June 2018).
18. *Autonomous Vehicles | Self-Driving Vehicles Enacted Legislation*; NCSL: Washington, DC, USA, 2019.
19. Madrigal, A.C. Inside Waymo's Secret World for Training Self-Driving Cars. *The Atlantic* 23 August 2017.
20. Dikmen, M.; Burns, C.M. 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*; ACM: New York, NY, USA, 2016; pp. 225–228.
21. Eustice, R. *University of Michigan's Work Toward Autonomous Cars*; Technical Report; University of Michigan: Ann Arbor, MI, USA, 2015.
22. Fagnant, D.J.; Kockelman, K. Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations. *Transp. Res. Part A Policy Pract.* **2015**, *77*, 167–181.
23. Autonomous connected cars on their way. *Shanghai Daily*, 8 June 2016.
24. Chang, L. Baidu partners with Shouqi Limo & Chauffeur to hasten autonomous car production. *MobileTech Daily*, 19 December 2017.
25. Kuang, X.; Zhao, F.; Hao, H.; Liu, Z. Intelligent connected vehicles: The industrial practices and impacts on automotive value-chains in China. *Asia Pac. Bus. Rev.* **2018**, *24*, 1–21.
26. Alibaba is developing its own self driving cars. *MIT Technology Review*, 16 April 2018.
27. Didi Chuxing gets permission to test self-driving cars in California. *CNBC*, 17 May 2018.
28. Self Driving Vehicles in an Urban Context. *SlideShare*, 24 November 2015.
29. Petit, J.; Shladover, S.E. Potential cyberattacks on automated vehicles. *IEEE Trans. Intell. Transp. Syst.* **2015**, *16*, 546–556.
30. He, Q.; Meng, X.; Qu, R. Survey on cyber security of CAV. In *Cooperative Positioning and Service (CPGPS)*; IEEE: Harbin, China, 2017; pp. 351–354.
31. Integrating Autonomous Vehicle Safety and Security, 2017. Available online: https://www.researchgate.net/publication/321323032_Integrating_Autonomous_Vehicle_Safety_and_Security (accessed on 10 March 2019).

32. Qayyum, A.; Usama, M.; Qadir, J.; Al-Fuqaha, A. Securing Connected Autonomous Vehicles: Challenges Posed by Adversarial Machine Learning and the Way Forward. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 998–1026.
33. Kumar, S.; Singh, K.; Kumar, S.; Kaiwartya, O.; Cao, Y.; Zhou, H. Delimitated Anti Jammer Scheme for Internet of Vehicle: Machine Learning based Security Approach. *IEEE Access* **2019**, *7*, 113311–113323, doi:10.1109/ACCESS.2019.2934632.
34. Cybersecurity Concerns with Self-Driving and Conventional Vehicles, 2017. Available online: http://umich.edu/~umtriswt/PDF/SWT-2017-3.pdf (accessed on 26 March 2019).
35. Cyber Security and Space Based Services—ESA Business Applications, 2019. Available online: https://business.esa.int/funding/invitation-to-tender/cyber-security-and-space-based-services (accessed on 31 May 2019).
36. Connected and Autonomous Vehicles: A Hacker's Delight? 2017. Available online: https://gowlingwlg.com/GowlingWLG/media/UK/pdf/autodrive/170907-cyber-security-white-paper.pdf (accessed on 28 March 2019).
37. Booch, G. *The Unified Modeling Language User Guide*; Pearson Education: New Delhi, India, 2005.
38. Ziegler, J.; Bender, P.; Schreiber, M.; Lategahn, H.; Strauss, T.; Stiller, C.; Dang, T.; Franke, U.; Appenrodt, N.; Keller, C.G.; et al. Making Bertha drive—An autonomous journey on a historic route. *IEEE Intell. Transp. Syst. Mag.* **2014**, *6*, 8–20.
39. Dolev, S.; Krzywiecki, L.; Panwar, N.; Segal, M. Certificating vehicle public key with vehicle attributes a (periodical) licensing routine, against man-in-the-middle attacks and beyond. In Proceedings of the SAFECOMP 2013-Workshop ASCoMS (Architecting Safety in Collaborative Mobile Systems) of the 32nd International Conference on Computer Safety, Reliability and Security, Toulouse, France, 27 September 2013.
40. The Newest Mobile Device: Self-driving Cars. *KWM*, 13 February 2018.
41. One autonomous car will use 4000 GB of data per day, 2018. Available online: https://www.networkworld.com/article/3147892/internet/one-autonomous-car-will-use-4000-gb-of-dataday.html (accessed on 15 June 2018).
42. Addressing Data Privacy Concerns in NHTSA V2V Rules. *Eversheds Sutherland*, 2 March 2017.
43. Hansman, S.; Hunt, R. A taxonomy of network and computer attacks. *Comput. Secur.* **2005**, *24*, 31–43.
44. Khurram, M.; Kumar, H.; Chandak, A.; Sarwade, V.; Arora, N.; Quach, T. Enhancing connected car adoption: Security and over the air update framework. In Proceedings of the 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT), Reston, VA, USA, 12–14 December 2016; IEEE: Piscataway, NJ, USA, 2016; pp. 194–198.
45. Ring, T. Connected cars–the next targe tfor hackers. *Netw. Secur.* **2015**, *2015*, 11–16.
46. Why the Next Denial-of-Service Attack Could Be Against Your Car. *IEEE Spectrum*, 28 October 2016.
47. Charette, R.N. This car runs on code. *IEEE Spectr.* **2009**, *46*, 3.
48. Zhao, L.; Kang, H.S.; Kim, S.R. Improved clustering for intrusion detection by principal component analysis with effective noise reduction. In *Information and Communication Technology-EurAsia Conference*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 490–495.
49. Tavallaee, M.; Bagheri, E.; Lu, W.; Ghorbani, A.A. A detailed analysis of the KDD CUP 99 data set. In *Computational Intelligence for Security and Defense Applications*; IEEE: Piscataway, NJ, USA, 2009; pp. 1–6.
50. Altwaijry, H.; Algarny, S. Bayesian based intrusion detection system. *J. King Saud Univ. Comput. Inf. Sci.* **2012**, *24*, 1–6.
51. Arora, I.S.; Bhatia, G.K.; Singh, A.P. Comparative Analysis of Classification Algorithms on KDD'99 Data Set. *Int. J. Comput. Netw. Inf. Secur.* **2016**, *8*, 34.
52. Lee, J.H.; Lee, J.H.; Sohn, S.G.; Ryu, J.H.; Chung, T.M. Effective value of decision tree with KDD 99 intrusion detection datasets for intrusion detection system. In Proceedings of the ICACT 2008—10th International Conference on Advanced Communication Technology, Gangwon-Do, Korea, 17–20 February 2008; IEEE:Piscataway, NJ, USA, 2008; Volume 2, pp. 1170–1175.
53. *MIT Lincoln Laboratory: DARPA Intrusion Detection Evaluation*; MIT Lincoln Laboratory: Lexington, MA, USA, 1998.
54. Hall, M.; Frank, E.; Holmes, G.; Pfahringer, B.; Reutemann, P.; Witten, I.H. The WEKA data mining software: An update. *ACM SIGKDD Explor. Newsl.* **2009**, *11*, 10–18.
55. Bhargava, N.; Sharma, G.; Bhargava, R.; Mathuria, M. Decision tree analysis on j48 algorithm for data mining. *Proc. Int. J. Adv. Res. Comput. Sci. Softw. Eng.* **2013**, *3*.

56. Sudrajat, R.; Irianingsih, I.; Krisnawan, D. Analysis of data mining classification by comparison of C4. 5 and ID algorithms. In *IOP Conference Series: Materials Science and Engineering*; IOP Publishing: Bristol, UK, 2017; Volume 166; p. 012031.
57. Witten, I.H.; Frank, E.; Hall, M.A.; Pal, C.J. *Data Mining: Practical Machine Learning Tools and Techniques*; Morgan Kaufmann: Burlington, MA, USA, 2016.
58. Patil, T.R.; Sherekar, S. Performance analysis of Naive Bayes and J48 classification algorithm for data classification. *Int. J. Comput. Sci. Appl.* **2013**, *6*, 256–261.
59. Alam, F.; Pachauri, S. Detection using WEKA. *Adv. Comput. Sci. Technol.* **2017**, *10*, 1731–1743.
60. Amor, N.B.; Benferhat, S.; Elouedi, Z. Naive bayes vs decision trees in intrusion detection systems. In *Proceedings of the 2004 ACM Symposium on Applied Computing*; ACM: New York, NY, USA, 2004; pp. 420–424.