Research, Implementation and Security Analysis of Connected and Autonomous Vehicles (CAVs) using Machine Learning Algorithms

Capstone Project

Presented by

Jaskiran Kaur

University of Alberta
Master of Science in Internetworking
Edmonton, Canada

Supervisor
Sandeep Kaur

ACKNOWLEDGEMENT

To begin, I would like to express my heartiest gratitude to my mentor Sandeep Kaur for her unwavering support of my report and research, for her patience, inspiration, and passion, as well as for sharing her vast expertise with me. Her counsel aided me throughout the research and writing stages of this thesis. I could not have asked for a more capable and supportive advisor and mentor for my Capstone Project.

Apart from my mentor, my sincere gratitude is extended to Dr. Mike MacGregor for providing me with the opportunity to conduct this research in the first place. I am grateful to Shahnawaz Mir and Sharon Gannon, who have supported me since my first semester, answered my foolish questions, assisted me, and pushed me to succeed in this course, despite my lack of experience in the subject. I would also want to convey my gratefulness to all of my other MINT instructors. Without their consistent aid and encouragement, I would not have successfully completed my master's.

Finally, but definitely not least, I want to thank my family: my parents, Gurkirpal Singh and Gurjeet Kaur, for giving birth to me and spiritually supporting me throughout my life.

ABSTRACT

Efforts linked to connected and autonomous vehicles (CAVs) have exploded in the last several years, and they are already beginning to impact people's daily lives. A growing number of businesses and academic institutions have made public announcements about their CAV efforts, and a few have even begun conducting field experiments. Governments all around the globe have also put rules in place to help hasten the adoption of CAV technology. CAV cyber security has emerged as a major concern, adding significantly to the difficulties of implementing CAV. However, there is no globally accepted or acknowledged paradigm for CAV cyber security. A UML-based CAV cyber security framework is proposed in this research according to UK CAV cyber security standards. It is based on this framework that possible CAV vulnerabilities are classified. Based on the highly tested KDD99 benchmark data set, a new CAV cyber-attack data set (called CAV-K99) is created. Communication-based cyberattacks against CAVs are the focus of this data set. Two machine learning techniques, Decision Tree and Naive Bayes, are used to create two classification models based on the CAV-K99 training data set. Comparisons are made regarding the models' accuracy, precision, and runtime for each sort of communication-based attack. The Decision Tree model has been proven to be better suitable for detecting CAV communication attacks and has a shorter runtime.

TABLE OF CONTENTS

LIST OF TABLES

1. Introduction

1.1 A new study area, connected and autonomous vehicles (CAVs), has recently gained much attention [1]. CAVs are becoming increasingly popular in both study and experimentation. In 2015 [2], the government of the United Kingdom established a government centre known as the "Centre for Connected and Autonomous Vehicles". In 2018 [3], this centre produced a report on research and development efforts involving connected and autonomous vehicles. In 2017 [4], the House of Lords produced a study titled "Connected and Autonomous Vehicles: The Future". The British Standard Institution (BSI), including many other enterprises, in the United Kingdom, produced a standard strategy report on CAVs in 2017 [5].

"Connected and 'Automated' Vehicles" has also been used in several publications. The Transport Systems Catapult [6], a UK-based innovation hub, uses the word 'Automated' on its website. CAV naming is inconsistent in the literature because it is still a growing paradigm. As a result, we use the phrase 'Connected and Autonomous Vehicle' in this report, which is synonymous with 'Connected and Automated Vehicles' in the literature .

1.2 Wireless communication and automation are the main attributes that CAVs are said to have. To plan their routes and interact with nearby vehicles within a linked network, they rely on data from other vehicles or infrastructure. The term "full automation" refers to a vehicle's ability to perform all dynamic driving operations and emergency procedures automatically, in real-time, without the need for the driver's involvement [7].

The Society of Automotive Engineers (SAE) [8] has categorized vehicle automation into six levels based on a variety of factors, including the capacity to perform lateral and longitudinal driving tasks at the same time, the potential to perceive and respond to objects and events, the ability to bounce back from a system failure, and the operational design domain constraint. The driver's and CAV system's responsibilities vary depending on the level of automation. The table below summarizes the details of level zero to level five automation.

Table 1: Automation Levels defined by the Society of Automotive Engineers (SAE) [8]

Level 0	No Driving Automation	• The driver performs all vehicle motion control functions. • The driver is in charge of keeping an eye on the surroundings and reacting to them. • If a system failure occurs, it is the driver's responsibility to restore it. • At this level of automation, the operational design domain does not exist.

Level 1	Driver Assistance	• The driver and the system work together to complete the driving task. The system can only regulate motion in one of two directions: longitudinal or lateral. • The driver is in charge of keeping an eye on the surrounding objects and events. • If a system failure occurs, it is the driver's responsibility to recover. • At this automation level, the operational design domain is constrained.
Level 2	Partial Driving Automation	• The system can control both longitudinal and lateral motion at the same time. • The driver is responsible for keeping an eye on all of the items and occurrences in their direct proximity. • If a system failure occurs, it is the driver's responsibility to ensure that the system is restored. • At this degree of automation, there is a restricted operational design domain to work with.
Level 3	Conditional Driving Automation	• The system can control both longitudinal and lateral motion at the same time. • The system keeps track of the objects and events in its direct proximity and responds to them. • In the event of a system breakdown, the driver must be prepared to respond to system requests or even take over direct control of the cars. • At this degree of automation, there is a restricted operational design domain to work with.
Level 4	High Driving Automation	• The system can control both longitudinal and lateral motion at the same time. • The system keeps track of the objects and events in its direct proximity and responds to them. • If a system failure occurs, it is the system's responsibility to recover from the failure. • At this degree of automation, there is a restricted operational design domain to work with.

Level 5	Full Driving Automation	• The system can control both longitudinal and lateral motion at the same time. • The system keeps track of the objects and events in its direct proximity and responds to them. • If a breakdown occurs, it is the system's responsibility to recover from the failure. • At this level of automation, the operational design domain is virtually limitless.

However, because of the unique characteristics of connectivity and autonomy, CAVs might be a little more sensitive to cyber-attacks and, therefore, more vulnerable while communicating information with their surroundings and other cars on the road [9]. Cyber security protects a computer system's functionality from cyber-attacks, especially harm to its infrastructure, programming, and data [10]. Cyber security in CAVs ensures the safety of the CAV system against cyber-attacks that damage the CAV's functionalities.

Furthermore, cyber-attacks on CAVs might be carried out both physically and remotely in order to steal, change, or destroy data. CAVs, which are expected to be the biggest portable devices in the near future, may have severe repercussions in people's lives, including the loss of sensitive data and the risk for catastrophic physical damages. In early 2018, an Uber autonomous car collided with a biker during road testing [11].

Tesla vehicles have also been linked to deadly accidents in the United States [12] and China [13]. According to Tesla, the driver's hands weren't detected on the steering wheel for six seconds just before collision in the United States. Even though the autopilot system was already activated, the automobiles should be characterized as a driver aid system rather than a completely autonomous system, as per automation levels defined in Table 1.

White-hat hackers in the United States have already successfully targeted the Grand Cherokee, gained control of the car and manipulated its windows [14]. As a result, even in the early phases of development, there is a strong need to explore CAV cyber security vulnerabilities. The CAV Standard Report has rated cyber security dangers as 'Very High,' [15] and the UK government issued the CAV Cyber Security Principles [16] in August 2017. Because of the substantial influence of CAVs on people's everyday life, CAV cyber security should be given top priority and addressed as soon as possible. These issues fueled the investigation into the CAV cyber security framework presented in this study.

CAV developers encounter complex cyber security problems as they create their products. For starters, the peculiarities of CAV cyber security make it difficult to anticipate all possible attacks before they occur. All developers and users must be aware that they must continually react to unexpected threats, as attack patterns are always changing. Hackers only need to discover one vulnerability flaw to execute an attack, but defenders must analyze all possible threats to protect CAVs. Second, CAVs are made up of various components and functionalities. Even if one of them

fails, the system as a whole can fail. Vulnerability testing is challenging given the multiple functions that operate together in a dynamic CAV system. Third, the many sensors in CAVs, capture massive volumes of data, which is challenging to handle, leave alone the aspect that the data is collected in various formats. To make data processing easier, the format and structure of the data input should be compliant with CAV protocols. Finally, CAVs connect via Wifi, Dedicated Short-Range Communications (DSRC), and Bluetooth, among other wireless communication networks. As a result, preventing CAV cyber security attacks is more challenging than preventing problems in wired networks.

There have been considerations of possible cyber security concerns in CAVs in the existing literature and attempts to design applicable frameworks to handle them. Nevertheless, there is still a need for a generally adopted structured methodology in which all sorts and areas of CAV cyber-attacks can be reliably identified and characterized and efficiently prevented. I will be presenting a brief description of the current state of CAV cyber security progress in this research and will design a UML-based framework for CAV relying on the principles [16] of UK CAV cyber security.

The new approach allows for more in-depth evaluation of cyber security concerns in CAV systems. In addition, the intrusion detection standard data set KDD99 [17] yields a new data set named CAV-K99. The attacks that are not relevant to CAV and duplicated in KDD99 have been eliminated from this new data set. The CAV-K99 data set includes 14 communication-based sub-attacks in CAV. Two machine learning algorithms, Decision Tree and Naive Bayes, are examined on the new data set and their efficiency, precision, and runtime is compared. Both methods are found to have equal accuracy, with Decision Tree having a faster runtime. Both systems, however, perform badly when it comes to identifying threats that aren't visible.

This is an intriguing topic for further research. The rest of this research is laid out as follows: A brief review of the relevant work on CAV cyber security is presented in Section 2. Section 3 uses UML to establish the interactions between elements in the CAV framework, and every class is explained in depth. Based on the latest CAV cyber security UML architecture, potential attack areas for CAVs are also described. In Section 4, the overlapping sorts of cyber-attacks in the benchmark data set, KDD99, are eliminated according to the new CAV framework. Section 5 statistically analyses the newly generated data set, named CAV-K99, predicated on that CAV framework. Two classification models are created using machine learning methods, and their performance in diagnosing CAV cyber-attacks is evaluated in terms of latency, accuracy, and precision. Section 6 brings the discussion to a conclusion by outlining my suggestions for further research.

2. Associated Cyber Security Work on CAV

As CAV's, being a relatively new study area, have lately garnered considerable attention throughout the world. Governments, businesses, research institutes, the media, and the general public have all been emphasizing on the development of CAVs, and some breakthrough has already been achieved. Some states in the USA have already enacted legislation permitting CAV road tests. [18]

Google [19] began developing self-driving cars in 2009, formed its subsidiary firm Waymo in 2016, and began a pilot program in Phoenix in 2018 to allow a small number of users to request driverless trips (where there is still a supervising driver in the vehicle for safety). Tesla [20] has been testing self-driving car technology on the road and commercializing it. Many papers have been published by colleges and universities in the United States, including the University of Michigan [21], with a Mcity test zone nearby.

Traditional leading automobile manufacturers in Europe, including BMW, Audi, and Mercedes Benz, have all made significant investments in CAV development [22]. Shanghai [23] was the location of the very first CAV test field in China. Baidu's Apollo CAV platform has been created, to produce Level 3 autonomous cars by 2019 [24]. Changan, BYD, Guangzhou Automobile Group, and Shanghai Automotive Industry Corp, all traditional automobile companies, have declared their CAV development ambitions [25].

Every year, a communication platform for the practical application of CAVs is offered at a CAV competition between universities [19] in China, on which the drawbacks and strengths of CAV may be identified, thereby adding to CAV research. Alibaba [26] and Didi Chuxing [27], both IT businesses, have entered this competitive arena. Furthermore, every day, the public witnesses scientific progress posted on websites and newspapers. People are eager to try and buy CAVs with 55 percent saying they would want to ride in a fully autonomous CAV as analyzed through a survey [28] conducted by the Boston Consulting Group. The vast majority of them would be willing to pay more than five thousand dollars on CAV functionalities in their vehicles.

Despite the significant efforts and resources made in the study and development of CAVs, there has been a disproportionately small amount of attention paid to the confidentiality and protection of CAV data. CAV cyber security is a subject with just a few books in the literature that are particularly related with it. Some preliminary efforts have been made to consider the possible attacks against CAVs, but they have been unsuccessful. A list of potential CAV cyber-attacks is provided in [29]. GNSS spoofing and the injecting of fraudulent messages were found amongst the most hazardous cyber risks, according to the analysis.

Potential cyber-attacks were divided into two categories in [30], passive and active. Active attacks, such as modification and spoofing, are easy to recognize but difficult to defend against because attackers can modify or fake the messages in the data transmission. Passive attacks, such as eavesdropping and information release, are challenging to detect but easy to defend against because the attackers do not interact with the data.

The authors of [31] pointed out that the present automotive safety standard ISO26262 does not take security into account to prevent both unintended and planned assaults. There is no common security or safety standard in place for CAVs at the moment. The development of CAVs would benefit greatly from a systematic specification of attacks and attack analysis tools. Other studies have explored particular assaults on CAVs, to suggest viable remedies utilising artificial intelligence, in addition to considerations of hypothetical attacks on CAVs. The authors of [32] conducted a thorough study of existing adversarial assaults on CAVs utilizing machine learning methods.

Potential attacks were further classified as the application layer, network layer, system-level, privacy breaches, sensor assaults, and so on. The authors of [32] stressed the need of intrusion detection in CAV development. The authors of [33] developed a system to forecast the location and identify the jamming attacks using the machine learning technique CatBoost and a Morsel supple filter.

The efficiency of vehicular communication has increased with maximum contribution from the anti-jamming method, which has improved accuracy and reduced packet loss ratio. The machine learning-based technique was shown to be successful in defending the CAV site from jamming assaults. CAV cyber-attacks might inflict physical harm to users, according to the preceding literature, unlike cyber security in other domains such as mobile devices. According to a University of Michigan poll [34], the public is more worried about the physical damage done by CAVs than the loss of sensitive data. However, there isn't enough relevant study on CAV cyber security, according to the findings. The European Space Agency (ESA) has issued a request for ideas for CAV cyber security systems that use artificial intelligence [35].

The following is a summary of the present research's discovered gap:

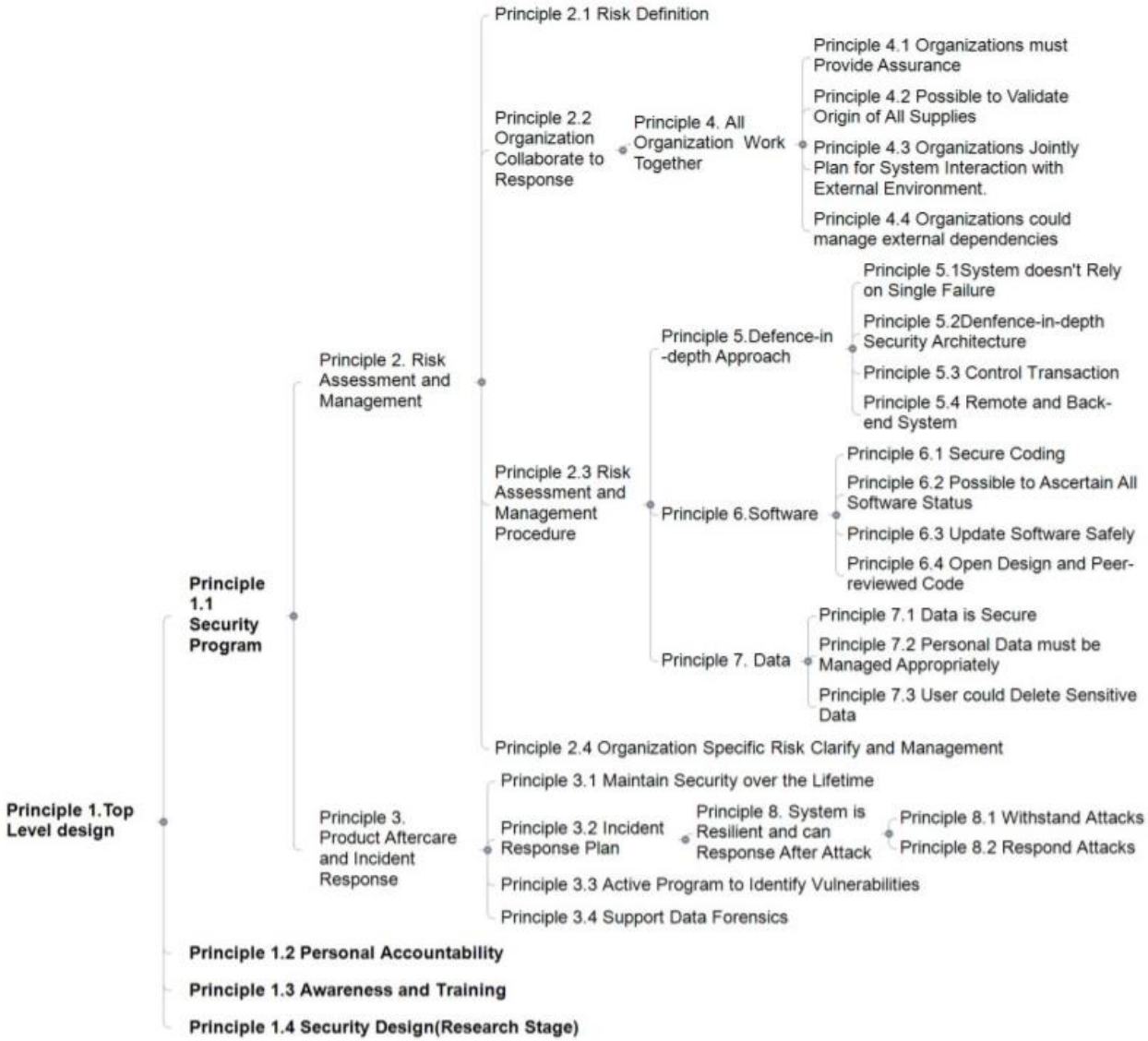
To begin with, there is no way to systematically analyze CAV vulnerabilities. Most of the research has been on specific attacks on CAVs, such as location spoofing or adversarial attacks on CAV algorithms.
It's also worth mentioning that there aren't many CAV cyber security data sets available because most studies have concentrated on theoretical elements, leaving detection tools in the dark. A systematic strategy for defining possible threats and establishing CAV cyber security data sets is required to address this vital research area in both business and academia.

In this study, a UML-based CAV framework is developed to analyse possible cyber security risks to CAVs, based on the UK CAV cyber security framework, to aid in the creation of a systematic approach for safeguarding CAV systems and data exchanged. A new data set, CAV-K99, was created for CAV cyber security detection. In order to assess their effectiveness in identifying CAV cyber security threats, two machine learning models based on Decision Tree and Naive Bayes are created.

3. UML-Based CAV Cyber Security Framework

The UK government issued a paper titled "Key Principles of Vehicle Cyber Security for Connected and Automated Vehicles" [16] in June 2017. The UK government issued eight CAV cyber security principles in this paper, which cover the whole life cycle of CAVs and provide protective guidelines to subcontractors, suppliers, and possible third parties in terms of hardware, software, and data. These eight principles, whose structure is seen in Figure 1, are summarised and categorized in this research.

Figure 1: Framework of cyber security guidelines for connected and autonomous vehicles (CAVs) based in the UK. [16]



Principle 1 is the most significant, as shown in Figure 1 since it outlines the top-level design criteria for CAV cyber security. Principle 1.4 (which analyses security program design) is an important step toward comprehensive protection, in combination with Principles 1.2 and 1.3, including human elements. In terms of the security program, Principle 1.1 splits the protection procedure into three stages:

1. Before the occurrence of the attacks: Relevant organizations, and manufacturers must specify the types of attacks that might occur, as well as the strategies for mitigating them.

2. When assaults occur: The system should keep an eye on the whole CAV and identify attacks as quickly as feasible. The system must also be able to withstand assaults.

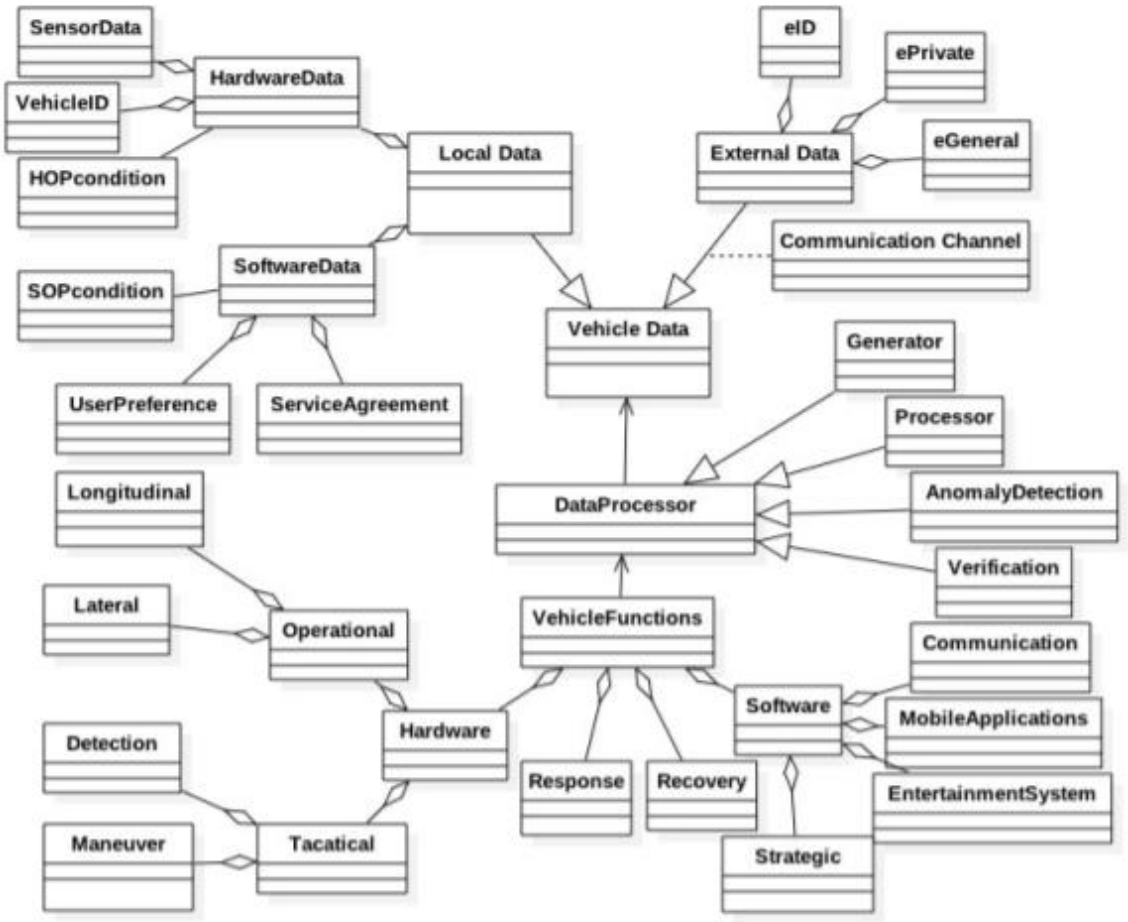
3. After an attack occurs: The system should respond correctly to attacks and be able to recuperate from them.

There is no generally adopted framework for CAV cyber security [36] in the literature, which may be used to define attack points and build effectivdefense solutions. The essential parts of CAV cyber security, according to the UK CAV cyber security principles we defined in Figure 1, are the defense-in-depth strategy, which covers physical, technical, and admin controls (Principle 5), software (Principle 6) and data (Principle 7). The risks of the CAV system may be specified, analyzed, and addressed before cyber security threats occur (Principles 2.1 and 2.3). Monitoring the CAV system during operations can assist in maintaining security throughout its lifespan (Principles 3.1 and 3.3). Following an assault, the CAV system may respond to and assist successful responses (Principles 3.2 and 8).

As a result, CAV cyber security may be separated into three categories: hardware, software, and data. CAVs create hardware, software, and data, but they are nevertheless connected to the outside world via data exchanges with other cars, infrastructure, and pedestrians, making the communication channel a vulnerability as well. The connections between these elements must also be specified.

In software engineering, the Unified Modelling Language (UML) is frequently used to create and represent system architectures [37]. A class diagram is used in UML to design a system's conceptual structure, displaying both the system's core components and their interactions with other components.

Figure 2: A CAV framework based on the Unified Modeling Language (UML). [37]



The suggested UML-based CAV cyber security framework, as illustrated in Figure 2, is designed to specify the interactions between each component and architecture in the CAV, including hardware, software, and their produced data, in order to aid the vehicle's proper operation. Different sorts and points of possible CAV cyber-attacks may be analysed and classified using the framework. Vehicle Data, Data Processor, and Vehicle Functions are the key classes in this UML-based CAV system.

3.1. Vehicle Data

In Cav's, Vehicle Data use data to make judgments and perform appropriate vehicle functionalities. As a result, Vehicle Data is the most important part of the CAV system. The information in the Vehicle Data class may be separated into two categories: local data and external data. The Vehicle Data class relates to Figure 1's Principles 5–7.

In the CAV framework, Local Data contains two sub-classes: hardware data and software data. These two sub-classes comprise not just data created by hardware and software, but also data about the hardware and software's functioning conditions. The HardwareData class contains sensor data acquired from the vehicle's surroundings by different CAV sensors, such as radar, GNSS, and camera [38]; for instance, GNSS and image data used to identify a CAV's current position. The VehicleID class also holds data that identifies the car, such as the license plate number (a unique registration number or letters assigned by the government). Since CAVs transfer data and information with several other entities, such as other CAVs, infrastructure and services, and pedestrians, VehicleID also contains a special pair, the public and private keys, used to encode and decode messages and check vehicle identification [39]. Hardware's operation condition data is stored in the HOPcondition class.

Local Data acquired by software in CAVs, such as the onboard entertainment system, is stored in the SoftwareData class. CAVs will very certainly become a popular smart mobile gadget in the future [40]. They not only offer decision-making assistance or solutions, such as the quickest driving route from point A to point B, but also cater to users' preferences, such as 'the most scenic route' or 'the peaceful route.' The UserPreference class holds user preference data, which CAVs take into account when making the optimal decision for individual users. The ServiceAgreement class specifies the conventions that the programme must follow, such as privacy protection and other service protocols. The SOPcondition class holds information about the software's operating conditions.

The ExternalData class contains data received from other entities in the communication system, such as other CAVs and intelligence infrastructures. All data is received via communication channels like Vehicle to Vehicle (V2V) and Vehicle to Infrastructure (V2I), which are part of the Communication Channel class. Because each entity has its own ID stored in its local data, the external data also need this information to ensure data sender identification, while the eID class holds the sender's ID information. Following the identification of the sender ID, messages in the external data are classified as either private or general, in accordance with Principle 7.2, which stipulates that data should be maintained responsibly. Vehicles or infrastructures may be required to convey private data, such as user preferences, under certain instances, which is kept in the ePrivate class and can only be viewed by authorized users. The eGeneral class maintains data that is accessible to everyone, such as vehicle size and position.

3.2. Data Processor

CAVs deal with large volumes of data each day. Each CAV is said to create up to 4000 GB of data in only one hour of driving [41]. Furthermore, adding a V2V communication network to a vehicle may necessitate 10 messages per second [42], increasing the data processors' strain. Even more essential than how the data is acquired is how it is handled. A data processor is incorporated into CAVs to clean data and assist in making suitable decisions. Figure 1 shows how the Data Processor class relates to Principles 2.3, 3.1, and 3.3.

The DataProcessor class has four fundamental data processing methods and is dependent on the Vehicle Data class. The Generator class collects data from several sources, necessitating the formats from numerous data sources to be controlled and fused in order to be processed. The Processor class cleans and annotates data in preparation for analysis. The Verification class contains components that ensure the data is safe, ensuring that the CAV system's cyber security standards are met. During these processing steps, the CAV system should be able to recognize aberrant circumstances in the hardware, software, and data. The AnomalyDetection class in the CAV system identifies any such flaws and abnormalities.

3.3. Vehicle Functions

If the CAV system is not behaving abnormally, relevant data will be utilized to make choices using the Vehicle Functions class once it has been processed. Vehicle Functions class is connected to Figure 1's Principles 3.1, 3.2, 5, 6, and 8, and is defined accordingly as illustrated in Figure 2. In the CAV framework, the functions of CAVs may be split into Hardware and Software classes, as illustrated in Figure 2.

The vehicle's dynamic driving activities and functions are classified into three categories with respect to SAE J3016 [8], namely operational functions, tactical functions, and strategic functions, with the former two falling under the Hardware category and the latter falling under the Software category. The standard vehicle motions, such as longitudinal and lateral movements, are included in operational functions. Tactical functions are in charge of monitoring the environment and the reactions that come with it and manoeuvre planning. There may be some overlap between operational and tactical tasks. Route planning is an important part of strategic operations. Strategic operations are excluded from the dynamic driving activities for now in the J3016 categories.

When a CAV senses items in its environment, it responds by using operational functions. The Hardware class can be classified into Operational and Tactical classes based on SAE J3016. The Operational class is divided into two sub-classes: longitudinal and lateral. When the vehicle moves longitudinally or laterally, these two sub-classes incorporate important hardware functions. There are two sub-classes in the Tactical class: The Detection class is used to track nearby objects and events using sensors such as radar, LiDAR, and cameras. The Manoeuvre class is responsible for performing necessary motions such as turning on the indicators.

Software features such as entertainment systems and mobile application capabilities and hardware functions are critical components of CAVs, which is why t he entertainment system and mobile apps are included in the Software class. The Communication class also includes support for all data receiving and transmitting capabilities. The Strategic class, which is described based on the strategic functions in SAE J3016, arranges the whole journey, including the optimum route, travel time, and destinations.

The Response class, in combination to the Hardware and Software classes, performs appropriate actions depending on the data from the hardware and software. The Recovery class is used to ensure that CAVs are robust and fail-safe in the event of a system breakdown.

3.4. Possible Attack Points

Viruses, worms, buffer overflows, DoS attacks, network assaults, physical attacks, password attacks, and information collecting attacks are all categories of cyber-attacks in computer networks [43]. Attacks on the stereo system or smartphone devices, including attacks on the Controller Area Network (CAN), which is an interior vehicle communication network for microcontrollers and gadgets, have been classified into two categories in typical car vehicles [44]. The second form of assault is more serious than the first since the CAN is linked to all of the in-vehicle hardware components, including the brakes, air conditioning, steering, and wheels.

CAVs, unlike computer networks and regular automobiles, are equipped with both physical and software pieces and linked to the whole transportation infrastructure. As a result, any of the previous attacks on automobiles might occur in a CAV. Furthermore, as the quantity of autonomy and networking functions grows, the number of vulnerabilities and attack opportunities will increase. CAV cyber security is required to defend the system from cyber-attacks that might damage its functioning remotely or physically. At an early stage, it is vital to identify, describe, and categorise probable forms of assaults against CAVs. The four categories of probable CAV assaults and sub-attacks are mentioned below, based on the UML-based CAV architecture shown in Figure 2.

3.4.1. Physical components of the vehicle.

The windshield, wheels, and even brakes are examples of CAV physical elements. Hackers have previously been claimed to be able to manipulate the brakes and air conditioners of Nissan [45] and JEEP automobiles. Due to this form of attack, JEEP recalled over 1.4 million cars to apply security fixes [46]. Attacks against hardware may be carried out either physically or remotely. The attack techniques include deceiving the hardware into making poor driving judgments or hacking into the hardware to listen in on conversations. They are enumerable attack points on the Cav's hardware.

Cameras, Light Detection and Ranging (LiDAR), and radars are among the most common sensors found on CAVs, as shown in Table 2. All of these sensors might be physically or remotely targeted; for example, faked visuals could fool the cameras, and the radar transmission could be blocked. Attackers might even get access to the vehicle's camera system in order to watch its operations. Furthermore, the GNSS system might be targeted by skilled attackers. The GNSS system, for example, might be jammed, preventing the vehicle from receiving a GNSS signal for navigation or positioning.

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, user´s 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).

3.4.2. Vehicle software.

Boeing's new 787 dreamliners are equipped with just 6.5 million lines of code [47], while CAVs might be loaded with more than 100 million lines of code. As a result, CAVs have a greater number of vulnerabilities. The entertainment system, the mobile apps that have been loaded, and the audio system aboard all have the potential to be used as attack sites by cybercriminals. If software is taken control of, the data interchange might be tracked, and the hardware could even be damaged beyond repair.

3.4.3. Data.

The data held on CAVs is communicated amongst CAVs, to infrastructure, and to pedestrians and cyclists through wireless connections. Local auto data such as the vehicle ID containing the electronic plate or the vehicle model and personal data such as user preferences might be compromised in an attack, resulting in data leakage. Additional to this, since CAVs may be used to support payment services (for example, toll services), private data such as money transfers might be used as an attack vector against CAVs. Foreign data collected from other users within the communication range might include attack sites as well as internal data. Modification of communication data or the introduction of fraudulent messages may result in not just information leakage issues, but also traffic congestion and even accidents in certain cases.

3.4.4. Communication channel.

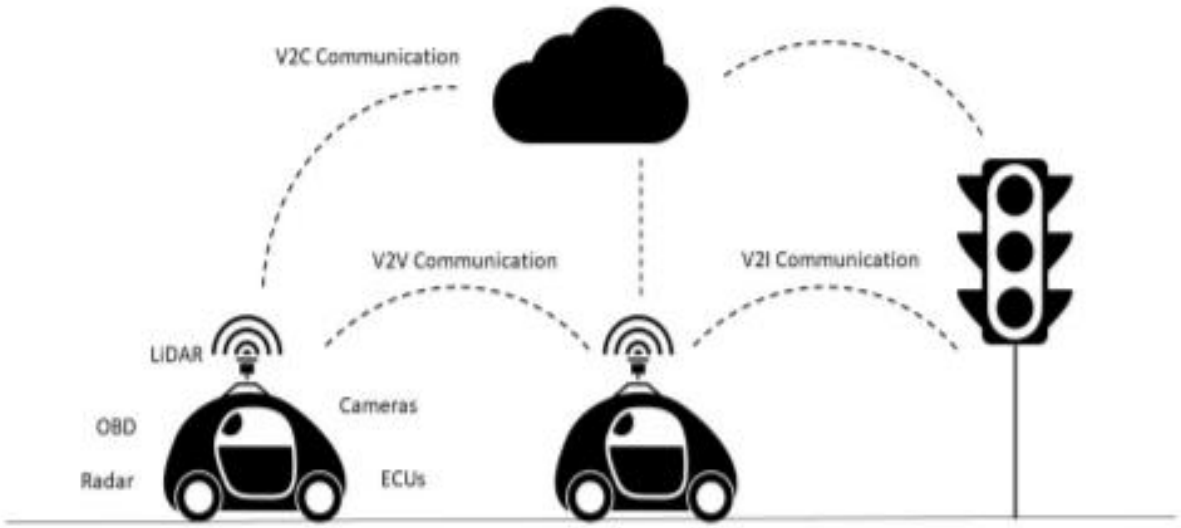
Potential assaults may also be directed against the routes of communication itself. V2V communication, Vehicle to Infrastructure (V2I) communication, Vehicle to Cloud (V2C) communication, and/or Vehicle to Everything (V2X) communication are all examples of attack

targets. If an attacker sends a large number of messages at the same time, a communication channel may be readily blocked by the network's defenses. In addition, eavesdropping on communication lines might result in the release of sensitive information.

These assessments have resulted in a list of probable attack areas for CAVs, which is shown in Table 2. Because the technologies used to combat CAVs are still in the early stages of development, the number of attack points will undoubtedly rise in the future. Nevertheless, since the attack sites are contained within the scope of physical components as well as software, data and communication routes, the framework may be expanded to accommodate and categorize new forms of assaults as they emerge.

In this paper, the research focuses on structuring the United Kingdom Cyber Security Principles [16] in order to develop a CAV cyber security framework categorizing communication-based attacks made wirelessly through communication channels to CAVs, which is then used to build a CAV cyber security framework. The use of machine learning methods to categorize these cyber-attacks is then illustrated. In Figure 3, the attack sites are shown, where CAVs share data [48] with their surrounding environments via V2V, V2I, and V2C communication channels.

Figure 3: Sites of Attack (Attack Points) through Communication Channel.



4. CAV-K99, the new CAV Cyber-Attack Data Set

CAVs, which are still in the early stages of development, will not be able to drive safely on public roads until they have been completely developed. When searching for and obtaining well-processed and labelled data sets on CAVs in the current literature, it might be problematic, particularly when searching for and obtaining data on CAV cyber-attacks. Specifically, I have adapted the widely used KDD99 benchmark data set on network intrusion detection [49] and have created a CAV communication-based cyber-attack data set (named CAV-K99) depending on different types of CAV cyber-attacks as well as the UML-based CAV framework established in Section 3 to address CAV communication-based cyber attacks.

4.1 The KDD99 Data Set

When it comes to online intrusion or attack detection, the KDD99 data set is a well-known benchmark. Initially made accessible during the Third International Knowledge Discovery and Data Mining Tools Competition in 1999 [49], it has since gained widespread use. It comprises data from regular network connections as well as simulated attack or intrusion data collected in a military network environment (KDD99). For more than a decade, the data set has been the most extensively used intrusion detection data set in the scientific literature [50].

KDD99 has roughly 5 million data records, each of which contains 42 characteristics. The 42nd attribute is a label that may be either normal or malicious in nature. To accommodate individuals who find the original data set to be too large for data processing, KDD99 additionally offers a ten percent data set consisting of around 500 thousand data records for training and testing.

4.2. Attacks existing in KDD99

The assaults in KDD99 are divided into four primary classes with a total of 39 sub-attacks [51], which are as follows [52]:

4.2.1. PROBE, which is an abbreviation for Probing attacks. During this sort of attack, the system is being monitored or scanned for weaknesses in order to obtain data from the system. The sub-attacks of PROBE in KDD99 contain ipsweep, mscan, nmap, portsweep, saint, and satan.

4.2.2. DoS, which stands for Denial of Service, is another kind of assault. DoS attacks prevent normal usage or communication in a system from taking place by using all of the system's resources, resulting in the system or communication channel being unavailable for normal use or communication. A typical exploit would include sending a large volume of data in order to overwhelm the communication connection and system. DoS attacks in KDD99 comprise apache2, back, land, mailbomb, Neptune, pod, processtable, smurf, teardrop, and udpstorm, to name a few examples. DoS attacks in KDD99 comprise apache2, back, land, mailbomb, Neptune, pod, processtable, smurf, teardrop, and udpstorm, to name a few examples.

4.2.3. The User-to-Root (U2R) attacks are carried out with the goal of gaining access to superuser accounts by the perpetrators. They find weaknesses in the system and then use those vulnerabilities to obtain access to the system's core. The U2R attacks in KDD99 include buffer overflow, httptunnel, loadmodule, perl, ps, rootkit, sqlattack, and xterm.

4.2.4. R2L, abbreviated for Remote-to-Local attack. As the term implies, the attackers' goal is to obtain access to the system and transfer data packets over a remote connection to a local system. The attacker does not have access to a legitimate account in the system, but he or she may get local access to one by exploiting a vulnerability. This list includes ftp write, guess passwd, imap, multihop, named, phf, send mail, snmpgetattack, snmpguess, spy, warezclient and worm (in KDD99), as well as the command line options xlock and xsnoop (in XSnoop).

It is noteworthy that there are 39 sub-attacks in the four primary attacks; yet, only 22 sub-attacks were included in training data set, indicating that the training data set is inadequate. The remaining 17 assaults are only included in the testing set. By using these data sets to test and validate detection approaches, we may also determine the robustness of detection strategies, such as those proposed and tested in Section 5 (machine learning algorithms).

KDD99 is a comprehensive data collection that covers a wide range of attack types that may be used against computer networks. Despite this, owing to the particular properties of CAVs noted above, the data set cannot be utilised directly for CAV cyber security in its current form. Sections 3 and 4 of this study describe how we modified and processed the KDD99 data set by eliminating unnecessary attack types, using the CAV framework that was built as well as prospective attack locations found in Sections 3 and 4. Table 3 lists the many sorts of attacks that exist in KDD99 that could possibly occur in CAV.

Table 3: Possible Sub-attacks on CAVs in KDD99 Data Set

Attack Type	Possibility		Attack Type	Possibility	
PROBE	ipsweep	H	R2L	ftp_write	H
	mscan	P		guess_passwd	H
	nmap	H		imap	I
	portsweep	P		multihop	P
	saint	P		named	P
	satan	P		phf	I
DOS	apache2	P		sendmail	P
	back	P		snmpgetattack	P
	land	P		snmpguess	P
	mailbomb	H		warezclient	P
	neptune	H		warezmaster	P
	pod	H		worm	H
	processtable	P		xlock	P
	smurf	H		xsnoop	H
	teardrop	H		spy	P

	udpstorm	H
U2R	buffer_overflow	H
	httptunnel	H
	loadmodule	I
	perl	I
	ps	I
	rootkit	P
	sqlattack	P
	xterm	I

Table 3 categorises the probable forms of CAV cyber-attacks into three categories: H stands for High, P for Possible, and I for Irrelevant. After the data was processed, the total amount of CAV attack types was decreased from 39 to 14, with 19 kinds of viable CAV attacks and 6 types of irrelevant attacks being identified as a result. The following are the reasons for data processing based on the sorts of attacks that have been identified:

1. Some of the attacks lacked a clearly defined objective. Considering that the data comes from the KDD99 dataset, the definitions of assaults are based on the original descriptions of the attacks themselves. Data from the DARPA intrusion detection assessment data set, which was gathered by the MIT Lincoln Lab [53] was used to create the KDD99 data set, which was then retrieved and analyzed. In this section, all descriptions of the assaults are taken directly from the official description available on the MIT Lincoln Lab web site [54]. Some sub-attacks lacked precise definitions and, as a result, could not be categorised as type P cyber-attacks under the CAV classification system. The sort of attack they use might be altered if a clear description is established.

2. Certain threats do not fall under the scope of the CAV cybersecurity framework. In Section 3, a CAV framework based on UML is developed to identify the various data types that are used in CAV communication and functions. KDD99, on the other hand, is a data set on computer and network security, and its protocols are distinct from those used by CAVs. The attack 'land,' for example, appears exclusively in earlier TCP/IP protocols and can only be discovered on an outdated Linux operating system known as SunOS 4.1, according to KDD99 . If the protocol and environment are no longer active, the potential of this attack may also be eliminated. These forms of attacks did not fit within the CAV framework and were thus deleted from consideration.

3. Some attacks were incompatible with the CAV attack sites, which was a problem. Besides physically damaging a CAV system, attackers must first identify one of the susceptible areas (as stated by Section 3) in the system before launching their attack. It is possible that these attack spots are located in physical components, software, data, or communication channels.

Some attacks in KDD99 can only take place under specified circumstances and on specific platforms, and as a result, they are not relevant to the CAV attack zones. The likelihood of these attacks occurring in CAV is quite limited; for example, the apache2 attack can only occur in an Apache Web Server environment. If a CAV does not make use of the Apache Web Server, the attack will be unable to be carried out.

5. Experiments

Anomaly detection plays a vital role in the CAV framework that was developed in Section 3. Using Weka [55], two machine learning algorithms were designed to generate two classification models, Naive Bayes and Decision Tree, in order to detect anomalous behaviour in the data. The experiments were performed using a machine with an Intel Core i3, 3.70GHz processor, and a 64-bit Windows operating system. Weka is an open source data mining programme created by the University of Waikato that has been extensively used in business and research to perform analysis and construct machine learning models. It is available for free download from the Weka website.

5.1. Data Preprocessing Using the CAV-K99

The KDD99 data collection contains more than 4 million data records and is too large to be processed on a personal computer due to its large size. Particularly, the training data set, which included 10% of the KDD99 data set, was employed in this work. It was decided to create a new data set termed CAV-K99 after deleting duplicates and unnecessary attack types from the original data set. This data set was designed to be compatible with the new CAV cyber security framework, which was developed by CAV. Tables 4 and 5 show the quantity of normal data and attack data contained in both the training and testing data sets, respectively.

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

	10% KDD99 Data	**CAV-K99 Data**
Attacks	396,743	54,485
Normal	97,278	87,832
Total	494,021	142,317

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

	10% KDD99 Data	**CAV-K99 Data**
Attacks	250,436	23,348
Normal	60,593	47,913
Total	311,029	71,261

Also, Table 6 shows the quantity of each sub-attack category in the CAV-K99 training and testing sets.

Table 6: Quantity of types of sub-attacks in KDD99 and CAV-K99

	10% KDD99 Training Data Set	**CAV-K99 Training Data Set**	**10% KDD99 Testing Data Set**	**CAV-K99 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

5.1.1. Learning and Testing Weka Software

Then, the CAV-K99 data was preprocessed in Weka using the following procedures:
1. In Table 3, the standard assault and 14 sub-attacks were designated with numbers ranging from 0 to 14.
2. Because the data ranges of each characteristic in the CAV-K99 data set and its test dataset were distinct, certain continuous data, such as duration and src_bytes, were normalized to make them more similar. The normalization process was carried out using the Unsupervised-attribute-normalize method in Weka, with the value range being set at 0 to 20.

3. The data was then required to be discretized at this point. The normalized data was discretized using the unsupervised-attribute-discretize technique in the Weka programming language. The unsupervised-attribute-numerictonominal technique was used to classify additional category attribute data, such as protocol type and service.

4. The properties with a single value were removed from the list of available attributes. The variables num outbound cmd and is host login were used. These characteristics have no effect on the detection since they have remained constant throughout the process. As a result, there were 39 qualities remained in CAV-K99.

5.2 Methods of Experimentation

CAV cyber-attacks were classified and detected using the machine learning algorithms Naive Bayes and J48, which were developed at Weka and used to construct the two classification models Naive Bayes and Decision Tree to categorize and identify CAV cyber-attacks.

5.2.1 Comparison of Machine Learning Algorithms

One of the most often used categorization models, the Decision Tree, has a high degree of readability [56]. A tree of nodes and branches linked by unidirectional edges is one of the categorization models in use today. With each internal node (and each branch leading to child nodes) of the Decision Tree, each attribute represents a choice variable with regard to that attribute, and every branch reflects a decision made on that attribute, extending to the child nodes of various attribute values. The categorization is represented by the leaves of the tree (which does not have any branches or child nodes).

The C4.5 approach is used by the J48 algorithm in Weka to construct the decision tree. C4.5 carries out the classification by computing the information gain ratio of every attribute and selecting the characteristics with the highest information gain ratio as the root node of the classification. If you want to get exact results while calculating the information gain ratio, you should first compute the amount of entropy transported by a data set with potential distribution values V using Equation (1), which is as follows [57]:

$$Entropy(V) = -\sum_{i=1}^{n} pi.\log(pi) \quad (1)$$

where n is the number of data set partitions (classification labels) and pi denotes the proportion of the ith partition. Hence, Equation (2) can be used to determine the information gain:

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

where a being the attribute, $|V_j|$ denotes the number of distributions in partition j, and $|V|$ denotes the number of distributions in partition V. Equation (3) can be used to obtain the information gain ratio, as follows:

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

where in Equation (4), the intrinsic value (IV) is computed as follows:

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

Once this tree is constructed, each value of the attribute is represented by a branch of the tree, and the data is divided into distinct classes or tree leaves. Until the information gain ratio hits the benchmark [58], which is set to 0.25 by default in this experiment, the procedure will be repeated. The probable distribution values for the CAV-K99 data set are represented by the 39 characteristics. After computing the information gain of all the attributes, the attribute dst host srv serror, which had the maximum information gain, was selected as the root node of the hierarchy of attributes.

The Bayesian probability model was used to construct the Naive Bayes algorithm. According to this assumption, all of the attributes in the information are independent, which means that each attribute has no influence on the other attributes [59]. When using the Naive Bayes model, the conditional probabilities of classes are calculated. The class with the greatest probability is the prediction result [60]. The equation of Naive Bayes is written as follows in Equation (5): [61]:

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

where P(c|X) is the posterior probability of class c underneath the predictor variable X, where X is the data set of attributes x1, x2,..., xn, P(X|c) is the conditional probability of class of predictor variable X, P(c) is the prior probability of class c, and P(X) is the prior probability of predictors X. In CAV-K99, c denotes the identifier of normal or attack data, and X denotes the data set of 39 attributes that have been selected. The probability of each data point in the testing data set belonging to distinct labels are determined based on their properties in the testing data set. Each bit of information is then assigned to the label with the greatest probability.

5.3. Results of Experiment

As previously noted in Section 4, after analyzing the original KDD99 data, the amount of attack types in CAV-K99 was decreased to 14. For the detection models, we employed CAV-K99 to construct them, which were then examined on the CAV-K99 testing data set. In order to prevent the overfitting problem, the training data set is used to develop the model first, followed by 10-folds validation. Then perhaps the machine learning model is tested against the CAV-K99 testing data set to ensure that it is accurate. Table 7 compares Decision Tree and Naive Bayes network models in terms of their overall accuracy, run time and precision. In this research, the accuracy is defined as the proportion of correctly categorised attacks in terms of total number of classification attempts.

Table 7: Runtime and Accuracy of Naive Bayes and J48

	Accuracy on 10-Folds Validation (%)	Accuracy on the Testing Data Set (%)	Time to Build Model (s)	Time on the Testing Data Set (s)
Naïve Bayes	99.42	95.66	0.15	3.38
J48	99.80	97.04	2.42	0.94

As can be shown in Table 7, the Decision Tree model attained the highest accuracy of the two models tested, despite the fact that the runtime was variable. If you are driving in real time, particularly when CAVs are travelling at high speeds, time is of the essence, since a large distance of more than 30 metres can be covered in less than a second. Naive Bayes required more time to identify threats with almost the same accuracy as Decision Tree, and as a result, Decision Tree was more productive for CAV cyber security than Naive Bayes.

Aside from that, because of the unique properties of CAVs, the rate of false positive (FP) attacks categorization is an important statistic to use in evaluating the effectiveness of the models. It is possible that, in real-world scenarios, a machine learning model would classify attack data as "normal data," which will have life-threatening effects. Table 8 shows the false positive rate calculated on the basis of this information. Additionally, as indicated in Table 8, the accuracy of each model developed using the following Equation (6) was evaluated as well.

$$Precision = \frac{TP}{TP+FP} \quad (6)$$

In this example, it can be observed that the false positive rate was significantly lower with 10-folds cross validation since all of the attack types had been examined and trained. This was in contrast to the false positive rate on the testing data set. On the testing data set, both models had a comparable false positive rate, and both models had an accuracy of more than 94 percent, which is excellent (94.84 percent and 94.64 percent, respectively). Based on these findings the false positive rate for both models was found to be acceptable.

Table 8: Rate of False Positive for Naive Bayes and J48

	FP on 10-Folds Cross Validation (%)	FP on the Testing Data Set (%)	Precision on Testing Data Set (%)
Naïve Bayes	0.1	5.2	94.84
J48	0.1	5.6	94.64

Table 9: Accuracy and False Positive rates of sub-attack types gained by Naive Bayes and J48

			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

It can be shown in Table 9 that both machine learning classification models showed good accuracy when it came to recognizing CAV cyber-attacks, which is encouraging. The percentage of false positives was low across the board in all of the attack data. When it came to detecting the PROBE attacks, Naive Bayes performed exceptionally well, however Decision Tree did not perform as well when it came to identifying the ipsweep attacks. When it came to recognising Denial of Service (DoS) assaults, both models performed equally; however, when it came to detecting pod attacks, the performance of Decision Tree was significantly greater. Due to the fact that there were only a limited number of records of the U2R and R2L assaults in the training data sets, both models performed badly under these attacks. However, it can be shown that Naive Bayes was still capable of detecting 2.3 percent of guess passwd assaults, with an accuracy that was marginally greater than that of the Decision Tree model.

It is important to note that both machine learning algorithms scored badly on attack types that were only included in the testing data set, such as mailbomb, udpstorm, httptunnel, worm, and xsnoop, and that this was consistent for both methods. There was a zero accuracy for the detection of these five attack types, which means that none of them are or will be detected. This is owing to the fact that both, the Decision Tree and Naive Bayes models, are built using supervised learning and, as a result, are unable to detect previously undiscovered attack types when they are implemented. Further research about the development of classification models or clustering models for previously discovered sorts of attacks will continue to be a fascinating area of exploration for the future research.

Based on the findings, it can be concluded that Decision Tree outperformed the competition when it came to communication-based threats in the CAV environment. The Decision Tree model demonstrated excellent accuracy and precision in detecting the attack in a short period of time. Although both models produced inadequate results when predicting previously unknown attacks, it should be noted that further research into this area is required in future studies.

6. Summary and Future Work

CAV technologies have become more sophisticated and well-established at this point. CAVs are expected to be on the road for commercial purposes as early as 2025, according to current estimates. However, challenges in CAV cyber security have not received the attention they deserve in comparison to other CAV technologies, despite the fact that they are becoming increasingly vital and high priority in current CAV innovations. As a result of a cyber-attack on an autonomous vehicle (CAV), catastrophic repercussions may occur, including the exposure of personal information, as well as bodily injuries and even fatalities. The significance of CAV cyber security has indeed been emphasized repeatedly by enterprises and the government in the United Kingdom.

On the basis of the UK CAV Cyber Security Principles, we conducted an analysis of several forms of CAV communication-based cyber security assaults and developed a UML-based CAV framework with various components. As a result of using this CAV framework as a guide, potential CAV attack areas were identified and classified. A new data set, termed CAV-K99, was created based on the benchmark data set KDD99, which contains 10 percent of the total data. Based on the suggested CAV cyber security framework, the irrelevant attacks and undefined threats were excluded from the original KDD99 data set, resulting in 14 categories of CAV cyber-attacks in the CAV-K99 data set. The original KDD99 data set had 39 different types of cyber-attacks. Furthermore, a significant quantity of duplicated normal and attack data was deleted from the original KDD99 data set as well.

In order to determine the accuracy of CAV cyber-attack detection using the two classification models, the newly developed CAV-K99 data set was analyzed statistically using two machine learning techniques, namely the Naive Bayes and Decision Tree. When it came to recognizing PROBE attacks, Naive Bayes outperformed Decision Tree, whereas Decision Tree outperformed Naive Bayes when it came to identifying DoS attacks. When it came to identifying U2R and R2L assaults, both models performed mediocrely. However, both algorithms were equally accurate in detecting the 14 assaults, with Decision Tree taking somewhat less time to complete the task. Based on the findings, it was determined that the Decision Tree method was better suited for identifying CAV communication-based attacks.

It was discovered that the classification techniques did not perform well on new forms of CAV cyber-attacks that had not previously been seen, i.e., those that had not been included in the training data set. Both models also fared badly in terms of recognising U2R and R2L attacks. It is critical for CAVs to have high detection accuracy in order to operate safely on public roadway. The usage of feature selection techniques and hybrid approaches can be employed in the future work in order to increase accuracy even more while simultaneously decreasing runtime. Additionally, the integration of supervised and unsupervised machine learning methods can be examined in order to increase the accuracy of recognizing previously unidentified attacks. It is also possible to increase the performance index of classification models when dealing with diverse forms of information. Furthermore, the attacks addressed in this study were all communication-based attacks, rather than physical ones. As a result, the CAV-K99 data collection does not contain all of the acknowledged

attack types against CAV that have been identified. In addition, there is an imbalance between the different categories of data in the data set. Furthermore, the technology underlying CAVs are still in the early stages of development. When more powerful processing units are applied to CAVs, the computational capabilities may be boosted significantly. The detection and examination of physical cyber-attacks and new forms of abuses and the enhancement of detection machine learning techniques are all potential study subjects for CAVs in the future, and they represent other promising research directions.

7. References

[1] E. Guerra, "Planning for Cars That Drive Themselves: Metropolitan Planning Organizations, Regional Transportation Plans, and Autonomous Vehicles," *Journal of Planning Education and Research,* vol. 36, no. 2, pp. 210-224, 2015.

[2] "Centre for Connected and Autonomous Vehicles," [Online]. Available: https://www.gov.uk/government/organisations/centre-for-connected-and-autonomous-vehicles.

[3] Connected and autonomous vehicle research and development projects, Centre for Connected and Autonomous Vehicles, Jul 11, 2018.

[4] Science and Technology Select Committee, Connected and Autonomous Vehicles: The future?, Authority of the House of Lords, 2017.

[5] BSI Group, "Connected and autonomous vehicles: Getting the standards right," Nov 2018.

[6] Catapult- Connected Places, "Connected vehicle project could end motorway pileups," [Online]. Available: https://cp.catapult.org.uk/news/connected-and-autonomous-cav-vehicle-project-could-end-motorway-pileups/.

[7] Department for Transport, "The Pathway to Driverless Cars," UK Government, Feb 2015.

[8] S. Reports, "Taxonomy and Definitions for Terms Related to Driving Automation Systems for On-Road Motor Vehicles," SAE Mobilus, 2021 (Revised).

[9] S. a. W. P. a. W. K. a. M. J. Parkinson, "Cyber Threats Facing Autonomous and Connected Vehicles: Future Challenges," *IEEE Transactions on Intelligent Transportation Systems,* vol. 18, no. 11, pp. 2898-2915, 2017.

[10] D. Schatz, R. Bashroush and J. Wall, "Towards a More Representative Definition of Cyber Security," *The Journal of Digital Forensics, Security and Law,* vol. 2, 2017.

[11] S. L. a. J. C. Wong, "Self-driving Uber kills Arizona woman in first fatal crash involving pedestrian," San Francisco, Mar 2018.

[12] "Tesla in fatal California crash was on Autopilot," Mar 2018. [Online]. Available: https://www.bbc.com/news/world-us-canada-43604440.

[13] N. E. Boudette, "Autopilot Cited in Death of Chinese Tesla Driver," 14 Sep 2016. [Online]. Available: https://www.nytimes.com/2016/09/15/business/fatal-tesla-crash-in-china-involved-autopilot-government-tv-says.html.

[14] H. '. c. J. f. 1. m. a. a. d. i. i. a. ditch, 23 July 2015. [Online]. Available: https://www.independent.co.uk/news/science/hackers-remotely-carjack-jeep-from-10-miles-away-and-drive-it-into-ditch-10406554.html.

[15] Governers' Highway Safety Association, [Online]. Available: https://www.ghsa.org/sites/default/files/2018-08/Final_AVs2018.pdf.

[16] G. UK, "The key principles of vehicle cyber security for connected and automated vehicles," Centre for Connected and Autonomous Vehicles, 2017.

[17] Stephen D. Bay and Dennis F. Kibler and Michael J. Pazzani and Padhraic Smyth, Center for Machine Learning and Intelligent Systems, [Online]. Available: https://archive.ics.uci.edu/ml/datasets/kdd+cup+1999+data.

[18] N. C. o. S. Legislature, "ncsl.org," Lexis Nexis, 18 Feb 2020. [Online].

[19] A. C. Madrigal, "theatlantic.com," The Atlantic, 23 Aug 2017. [Online].

[20] M. Dikmen and C. M. Burns, "Autonomous Driving in the Real World: Experiences with Tesla Autopilot and Summon," Association for Computing Machinery, 2016.

[21] R. Eustice, *University of Michigan's Work Toward Autonomous Cars,* USA: University of Michigan.

[22] D. J. Fagnant and K. Kockelman, "PREPARING A NATION FOR AUTONOMOUS VEHICLES: OPPORTUNITIES, BARRIERS AND POLICY RECOMMENDATIONS FOR CAPITALIZING ON SELF-DRIVEN VEHICLES," Transportation Research, Austin, Texas, 2015.

[23] A. Lu, "ShanghaiDaily.com," Shine Beyond a Single Story, 08 Jun 2016. [Online].

[24] J. Warning, "Mobile World Live," Disqus, 30 Oct 2017. [Online].

[25] F. Z. H. H. &. Z. L. Xu Kuang, "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, 2017.

[26] G. Wang, "At the Forefront of Turning AI into Consumer-Ready products," MIT Technology Review, China, 2017.

[27] R. Browne, "CNBC," 17 May 2018. [Online]. Available: https://www.cnbc.com/2018/05/14/didi-chuxing-gets-permission-to-test-self-driving-cars-in-california.html.

[28] T. Litman, "Autonomous Vehicle Implementation Predictions, Implications for Transport Planning," Victoria Transport Policy Institute, Mar 3, 2022.

[29] J. Petit and S. E. Shladover, "Potential Cyberattacks on Automated Vehicles," *IEEE Transactions on Intelligent Transportation Systems,* vol. 16, no. 2, pp. 546-556, 2015.

[30] D. K. R. A. R. a. J. M. C. Bhisham Sharma, "Attacks to Automatous Vehicles: A Deep Learning Algorithm for Cybersecurity," *Sensors 2022,* p. 360, 22(1).

[31] Z. Hu, "Analysis of Autonomous Vehicle Safety Constraints Based on Systems-Theoretic Process Analysis," *Journal of Physics: Conference Series,* vol. 1650, 2020/10/01.

[32] M. U. J. Q. A. A.-F. Adnan Qayyum, "Securing Connected & Autonomous Vehicles: Challenges Posed by Adversarial Machine Learning and The Way Forward," *IEEE Communications Surveys and Tutorials 2020,* 2019.

[33] 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," Aug 14, 2019.

[34] N. Kowalskyj, "Cybersecurity and Autonomous Vehicles," Isabella Blandisi-Van Hee, 2020.

[35] M. Sivak and B. Schoettle, "Cyber Security Concerns with Self-Driving and Conventional Vehicles," University of Michigan Sustainable Worldwide Transportation, Michigan 48109-2150, USA, 2017.

[36] N. E. Vellinga, "Legal Aspects of Automated Driving: On Drivers, Producers, and Public Authorities," University of Groningen, 2020.

[37] G. Booch, J. Rumbaugh and I. Jacobson, The Unified Modeling Language User Guide, 1 ed., Addison Wesley, October 20, 1998, p. 512.

[38] J. Ziegler, T. Dang, U. Franke, H. Lategahn, P. Bender, M. Schreiber, T. Strauss, NilsAppenrodt, C. G. Keller, E. Kaus, C. Stiller and R. G. Herrtwich, "Making Bertha Drive — An Autonomous Journey on a Historic Route," *JOURNAL OF LATEX CLASS FILES, VOL. 11, NO. 4, DECEMBER 2012 2,* pp. 8-20, 2014.

[39] S. a. P. N. a. S. M. K. Ł. Dolev, "Certificating Vehicle Public Key with Vehicle Attributes: Against Man-in-the-Middle Attacks and Beyond," *A Licensing Routine,* 09 2019.

[40] M. Schaub, D. Hong and A. Zhao, "The Newest Mobile Device: Self-driving Cars," 13 Feb 2018. [Online]. Available: https://cmkwmlive.kwm.com/en/cn/knowledge/insights/the-newest-mobile-device-self-driving-cars-20180213.

[41] P. Nelson, "Just one autonomous car will use 4,000 GB of data/day," 07 Dec 2016. [Online]. Available: https://www.networkworld.com/article/3147892/one-autonomous-car-will-use-4000-gb-of-dataday.html.

[42] T. F. T. J. S. Mary Jane Wilson-Bilik, "Addressing V2V Data Privacy Concerns In New NHTSA Rules," Portfolio Media. Inc., New York, 2017, March 02.

[43] R. H. Simon Hansman, "A taxonomy of network and computer attacks," *Computers & Security,* 2005, Feb.

[44] H. K. A. C. V. S. N. A. T. Q. Muzaffar Khurram, "Enhancing connected car adoption: Security and over the air update framework," *IEEE 3rd World Forum on Internet of Things (WF-IoT),* pp. 194-198, 2016.

[45] T. Ring, "Connected cars – the next targe tfor hackers," *Network Security,* pp. 11-16, November 2015.

[46] T. S. PERRY, "Why the Next Denial-of-Service Attack Could Be Against Your Car," 28 Oct 2016. [Online]. Available: https://spectrum.ieee.org/why-the-next-denial-of-service-attack-could-be-against-your-car.

[47] R. N. Charette, "This Car Runs on Code," [Online]. Available: https://www.bu.edu/smartlighting/files/2010/01/IEEE-Spectrum_-This-Car-Runs-on-Code.pdf.

[48] Q. He, X. Meng, R. Q. 2 and R. X. 1, "Machine Learning-Based Detection for Cyber Security," Nottingham, Aug 2020.

[49] H.-S. K. S.-R. K. Lu Zhao, "Improved Clustering for Intrusion Detection by Principal Component Analysis with Effective Noise Reduction," in *1st International Conference on Information and Communication Technology (ICT-EurAsia)*, Yogyakarta, Indonesia, March 2013.

[50] E. B. W. L. a. A. A. G. Mahbod Tavallaee, "A Detailed Analysis of the KDD CUP 99 Data Set," in *Proceedings of 2009 IEEE Symposium on Computational Intelligent in Security and Defense Applications (CISDA 2009)*, 2009.

[51] S. A. Hesham Altwaijry, "Bayesian based intrusion detection system," *Journal of King Saud University - Computer and Information Sciences,* pp. 1-6, January 2012.

[52] I. S. Arora and Gurpriya Kaur Bhatia, "Comparative Analysis of Classification Algorithms on KDD‟99 Data Set," *I. J. Computer Network and Information Security,* pp. 34-40, 2016.

[53] J. L. S.-g. S. J.-h. R. T. M. C. Joong-Hee Lee, "Effective Value of Decision Tree with KDD 99 Intrusion Detection Datasets for Intrusion Detection System," in *10th International Conference on Advanced Communication Technology*, 2008.

[54] M. I. o. Technology, "1998 DARPA INTRUSION DETECTION EVALUATION DATASET," February 1998. [Online]. Available: https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset.

[55] 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, Jun 2009.

[56] G. S. R. B. M. M. N. Bhargava, "Decision Tree Analysis on J48 Algorithm for Data Mining," 2013.

[57] 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*, 2017.

[58] I. H. Witten, E. Frank and M. A. Hall, Data Mining Practical Machine Learning Tools and Techniques, Elsevier Inc..

[59] T. R. Patil and M. S. S. Sherekar, "Performance Analysis of Naive Bayes and J48 Classification Algorithm for Data Classification," *International Journal Of Computer Science And Applications,* April 2013.

[60] F. Alam and S. Pachauri, "Comparative Study of J48, Naive Bayes and One-R Classification Technique for Credit Card Fraud Detection using WEKA," *Advances in Computational Sciences and Technology,* vol. 10, no. 6, pp. 1731-1743, 2017.

[61] "Naive Bayes vs decision trees in intrusion detection systems," in *ACM DIgital Library*, March 2004.

[62] G. o. UK, Innovation is great: connected and automated vehicles, UK: HM Government, 2020.