# RSU-Based Intrusion Detection and Autonomous Intersection Response Systems

Peter Yurkovich

Thesis submitted to the Faculty of the

Virginia Polytechnic Institute and State University

in partial fulfillment of the requirements for the degree of

Master of Science

in

Civil Engineering

Kevin Heaslip, Chair

Alan Michaels

Hesham Rakha

Febuary 16, 2022

Blacksburg, Virginia

Keywords: Vehicle ad hoc Network, Intrusion Detection System, Intrusion Response System, Autonomous Intersection

# RSU-Based Intrusion Detection and Autonomous Intersection Response Systems

Peter Yurkovich

(ABSTRACT)

Vehicular safety and efficiency has been an ongoing research topic since the creation of the automobile. Despite this, deaths due to vehicular accidents are still extremely common, with driver issues and errors causing a vast majority of them. In order to combat the safety risks, Connected and Autonomous Vehicles (CAV) and other smart solutions have been heavily researched. CAVs provide the means to increase the safety of travel as well as its efficiency. However, before connected vehicles can be deployed and utilized, safe and secure communication and standards need to be created and evaluated to ensure that the introduction of a new safety threat does not overshadow the one that is already being faced. As such, it is integral for Intelligent Transportation Systems (ITS) to prevent, detect and respond to cyberattacks.

This research focuses on the detection and response of ITS components to cyberattacks. An Intrusion Detection System (IDS) located on Roadside Units (RSU) was developed to detect misbehavior nodes. This model maintains a 98%-100% accuracy while reducing system overhead by removing the need for edge or cloud computing. A resilient Intrusion Response System (IRS) for a autonomous intersection was developed to protect again sybil attacks. The IRS utilizes adaptive switching between several intersection types to reduce delay by up to 78% compared to intersections without these defenses.

# RSU-Based Intrusion Detection and Autonomous Intersection Response Systems

Peter Yurkovich

(GENERAL AUDIENCE ABSTRACT)

Vehicular safety and efficiency has been an ongoing research topic since the creation of the automobile. Despite this, deaths due to vehicular accidents are still extremely common, with driver issues and errors causing a vast majority of them. In order to combat the safety risks, Connected and Autonomous Vehicles (CAV) and other smart solutions have been heavily researched. CAVs provide the means to increase the safety of travel as well as its efficiency. However, before connected vehicles can be deployed and utilized, safe and secure communication and standards need to be created and evaluated to ensure that the introduction of a new safety threat does not overshadow the one that is already being faced. As such it is integral for Intelligent Transportation Systems (ITS) to prevent, detect and respond to cyberattacks.

This research focuses on the detection and response of ITS components to cyberattacks. An Intrusion Detection System (IDS) was created to detect vehicles misbehaving or conducting cyberattacks. The IDS is installed on off-road computers, called Roadside Units (RSU) which prevents the need for a separate server to be created to hold the IDS. The IDS is able to identify misbehavior and attacks at a 98% to 100% accuracy. An autonomous intersection is an intersection where all directions for driving through the intersection are transmitted through wireless communication. A Intrusion Response System (IRS) was developed for an autonomous intersection, to defend against vehicles making multiple reservation requests to

pass through the intersection. The IRS reduces vehicle delay through the intersection by 78% compared to an intersection without defenses.

# Acknowledgments

I would like to thank Dr. Kevin Heaslip for his help and support throughout my time at Virgnia Tech. His expertise, dedication and knowledge have all helped to shape me and the work that I have created.

I would like to thank my committee for their guidance throughout the creation and completion of my thesis.

My friends and family have supported me to no end, and I wouldn't have been able to complete my thesis without their unwavering support.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Intelligent Transportation Systems (ITS) and Connected and Autonomous Vehicles (CAV) have become one of the most studied areas within transportation. Despite vehicle safety increasing every year, in just the USA, over 35,000 people die every year in crashes, with an estimated 94% of crashes being caused by driver issues and errors [43, 44]. CAV and ITS have become some of the primary ways for researchers and designers to help minimize the driver issues and errors to help increase safety. The frequent sending of Basic Safety Messages (BSM) as well as situation specific messages can provide up to date and relevant information which might not have otherwise been available to CAVs. Infrastructure, such as Roadside Units (RSU), can be used to transmit safety critical data from a global perspective down to an individual vehicle. Vehicle Ad-hoc Networks (VANET) are utilized for transmitting data wirelessly between CAVs and between CAVs and RSUs. Alongside increasing safety, VANETs have also been researched to increase efficiency within vehicle, vehicle clusters, and in system-wide behavior.

Vehicle automation is often defined using the SAE automation levels [47]. Within this, automation levels of 4 and 5 are typically expected when discussing CAVs. Vehicles which are unable to communicate or do not fulfil the requirement of SAE automation levels are referred to as Legacy Vehicles (LV) [7], and may or may not be included within research of CAVs. The CAV market penetration, or percentage of vehicles which fulfil CAV requirements, is used when LVs are considered.

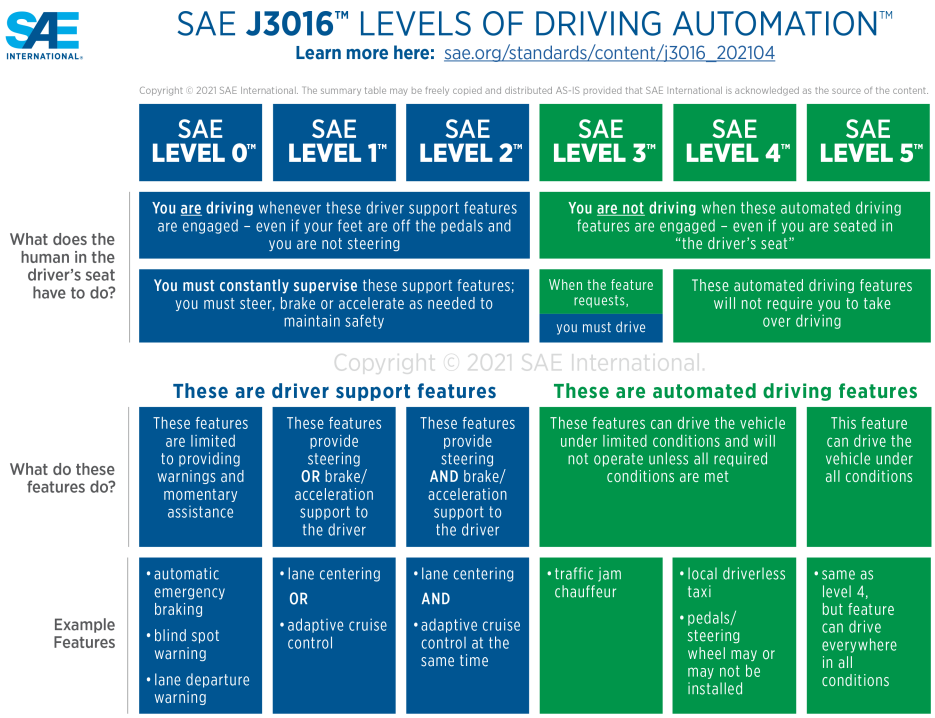



Figure 1.1: SAE Automation Levels [47]

However, before VANETs and the applications can be deployed, it is critical to create and provide safe, secure and reliable networks, protocols and security systems. The CIA triad of Confidentiality, Integrity and Availability, as well as Non-Repudiation and Authentication are often seen as the five critical security requirements for VANETs to be able to fulfil [1]. In order for a security system to be efficient, its ability to prevent, identify and respond to cyberattacks is critical [59].

## 1.1 Statement of Importance

This paper is constructed in the manuscript format, containing two projects. The first is an Intrusion Detection System (IDS) which monitors BSMs for malicious or faulty behavior.

The IDS is the first to be located and trained upon data available to RSUs within a standardized data set or framework. The IDS is able to use location specialization to increase IDS performance by reducing overall input complexity. Compared to other state of the art systems, it reduces system overhead, costs, data transmission, and improves privacy. This research was built upon the Framework for Misbehavior Detection [32]. The results show that the RSU-Based IDS is able to have great performance compared to current literature.

The second is an Intrusion Response System (IRS) located within an RSU which is running an autonomous intersection. This is the first paper which considers an IRS located at an RSU level. It is one of the first to discuss how the physical behavior of an Intelligent Transportation System should respond to cyberattacks. It introduces the concept of adaptively switching intersection types to minimize delays caused by cyberattacks. It also introduces the idea of a wireless signalized intersection and a wilfully compromised system. These concepts should be used in the design and creation of RSU and other VANET infrastructure to prevent significant impact from cyberattacks.

The rest of this thesis is structured as follows. Chapter 2. contains a Literature Review of VANET applications and security. This section provides an overview of concepts and implementations which are utilized in both papers. Chapter 3. introduces a Hierarchical Intrusion Detection System based within an RSU. Chapter 4. evaluates an Intrusion Response System for a Autonomous Intersection, and Chapter 5. contains conclusions, key findings, and future work. Appendix A contains further data for Chapter 3.

# Chapter 2

# Review of Literature

## 2.1 VANET Applications

Within VANETs, applications can be broken down into three distinct categories: safety, efficiency, and User-Based [1]. Safety applications are messages, protocols and programs which are run to increase the level of safety for users of CAVs. Efficiency applications have a wider range of applications under them, including intersection control, vehicle routing, message routing, and environmental impact reduction among other possibilities. User-Based applications are typically considered for behavior which falls outside of the other two categories. Personal data streaming, internet connection, games, and weather information would all fall into this category [46]. Some applications may fall into multiple categories, such as cooperative adaptive cruise control which uses communication to reduce distance between vehicles to increase efficiency, as well as disseminating information on stops and slowdowns to improve safety. Further details on safety and efficiency applications are included below.

### 2.1.1 Safety Applications

Safety applications for VANETs have been historically classified into five categories: collision avoidance, information forwarding, public safety, information exchange, and vehicle maintenance [71]. These categories can be further condensed down to Information Forwarding and

Local Information. Information Forwarding contains applications which sends data whose origin is not local to a vehicle, while Local Information applications send data originating from the vehicle.

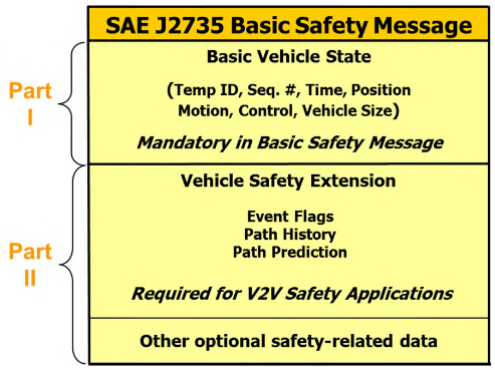



Figure 2.1: Basic Safety Message Structure [6]

The primary Local Information application is the periodic sending of BSMs. Each BSM holds safety and vehicle state data which is sent out often to allow vehicles to create maps of surrounding vehicles [54]. Each BSM is required to contain the time sent, position, position accuracy, speed, heading, acceleration, vehicle size, and status of the brake, transmission, and steering wheel systems. BSMs are expected to be transmitted at a frequency of 10 Hz so long as the network is able to sustain it. Since BSMs are seen as the information disseminator of choice within VANETs, many cyberattacks and defenses are targeted and trained at data within BSMs. As vehicles are expected to make decisions based upon the data within BSMs, modification of the data within a BSM could cause slowdowns or crashes. Other applications include context specific messages, such as lane change requests or warnings, forward collision warnings, blind spot warnings, and breaking warnings.

Information Forwarding applications consider information which originates from outside the vehicle. Within each application, information is retrieved from communications and external factors and then sent. Local Information, on the other hand, only requires sending informa-

tion since the information it retrieves is entirely internal. Traffic signal warnings, parking information, and emergency vehicle warning are all applications which fall under information forwarding. CAVs forwarding Local Information messages is another critical Information Forwarding application.

### 2.1.2 Efficiency Applications

Efficiency applications are a broad classification for applications within VANETs which seek to improve an aspect of the ITS system. Several of the areas which are improved upon are traffic and vehicle control, wireless communication routing, and energy usage. Energy reduction has been evaluated in a number of papers and has shown its effectiveness [76]. Wireless communication routing has also been heavily studied to determine efficient routing and network use while operating in VANETs. [55] evaluates traditional routing protocols such as AODV, DSDV, and OLSR within the context of VANETs. [36] utilizes Q-learning to reduce the delay, number of hops, and increases delivery the delivery ratio of messages. [74] utilizes RSU-assisted traffic-aware routing to skip over as many hops are possible between RSUs, while ensuring there are enough vehicles between the RSU and receiver to forward the message, through the use of Q-Learning.

The final set of efficiency applications are those which directly control a single vehicle's pathing or the entire traffic flow. Traffic flow control or path planning optimizes multiple path options for a number of vehicles requiring direction. [28] calculates optimal pathing for an entire district, and minimizes information gathering complexity through behavior prediction. [40] proposes a cooperative game hosted at RSUs to optimize pathing. [31] divides areas into grids to create a hierarchical model which reduces complexity within grids and reduces overall computational complexity. Individual vehicle control is often seen within

clustering approaches such as cooperative adaptive cruise control or when considering the control of an autonomous intersection.

Intersection management and control within a fully autonomous and wireless intersection is crucial to be completely safe and secure before any physical testing begins. Autonomous intersections can be broken up into two critical views, macroscopic and microscopic. A macroscopic view within traffic simulations often focuses on traffic flows as liquids, using flow, speed, and density. Instead the macroscopic focus in an autonomous intersection is the state of signal and network functioning. In the literature, the intersection management has been divided up into three potential states: Centralized Intersection Management, Distributed Intersection Management, and Wireless Intersection Failure [17].

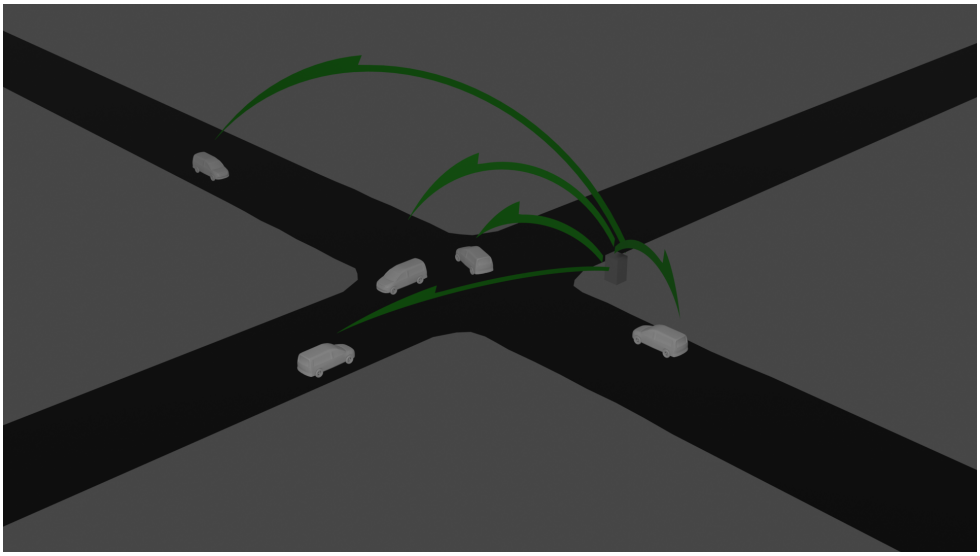


Figure 2.2: Centralized Intersection Management Format

Centralized Intersection Management (CIM) follows a fairly set procedure. A vehicle will send a request to pass through the intersection to an RSU. The RSU will make reservation decisions for each vehicle requesting, and may change previous reservations based upon optimization choices. The RSU will then send out responses to vehicles requesting, as well as to any vehicles whose reservation has changed [73]. Within the literature we see examples of

optimization [73], multi-agent approaches, intersection as a market [21], trajectory planning [75], and biologically inspired decision making [11].

Distributed Intersection Management (DIM) follows most of the same procedure as CIM, but has the additional need to dictate a vehicle or group of vehicles as decision makers. A single vehicle can be elected as controller based upon trust factors and availability of computational power and would then pass the controller role off to the next chosen controller as it leaves the intersection. A priority based approach which utilizes blockchain as the basis for decision making has been proposed and shown to maintain adequate send rate and feature an increase of 5% in average speed compared to several popular alternatives [12]. A hierarchical approach has been proposed which was able to decrease average time to cross the intersection and fuel consumption per car [66].

Wireless Intersection Failure (WIF) has been substantially less studied. Wireless Intersection Failure is often shown through the intersection reverting to function as a four way stop, which can be extremely inefficient at any substantial vehicle flow rate. More detailed information on CIM, DIM and WIF will be provided in Chapter 4.

## 2.2 VANET Security

Vehicle Ad-hoc Networks deal with nodes moving at high speeds, strict security requirements, potentially limited computing power, and high risks for the physical component of the cyber-physical system. VANETs are complex and difficult systems to secure and design for. The previously utilized Wireless Access in Vehicular Environments (WAVE) protocol set the requirements for secure VANET systems [27]. Despite the communication technology underneath it being outdated, the abstraction of the security requirements set forth still remain valid. Security Credential Management System (SCMS) was created as the security

system to be utilized within VANETs, and looks to fulfil the abstractions set forth in the WAVE protocol [10].

### 2.2.1 Security Requirements

Security requirements within communication are often summarized with the CIA Triad of Confidentiality, Integrity and Availability. Authentication and non-repudiation are often added into the CIA Triad to round out the security requirements for a communication system [1].

Confidentiality within cybersecurity indicates that unknowing parties are not able to know who is sending the data and what data is being sent. Confidentiality ensures that only authorized users and entities will have access to the data that is being sent. Confidentiality within VANETs is especially important to ensure privacy of message senders and prevent eavesdropping. Without confidentiality within VANETs, it becomes trivial to track vehicle movements. Within VANETs, the usage of pseudonyms is a common concept for users to be able to send data without the worry of losing their privacy. While pseudonyms are not able to protect a users' privacy from the systems providing the pseudonyms, it will protect the user's privacy from other normal users.

Integrity within cybersecurity ensures that the data that is being sent or stored remains constant. Integrity ensures that messages being sent are not able to be modified unknowingly, and that the data being sent to a third party is the same data that the third party is receiving. Within VANETs, because messages can often be safety-critical, it is important to ensure that the messages that are being sent are being received in the exact same form. The three most common attacks on data integrity within VANETs are message manipulation, the man-in-the-middle (MITM) attack, and a replay attack. Message manipulation can include bit

flipping, full randomization of messages, or modification of specific parts of a message. A replay attack is simply resending a message that an attacker received at a later time, which could cause a vehicle to act inappropriately for the real situation around it. A MITM attack will look to intercept messages between two real users, authenticate with one or both of the users, and then serve as a intermediary or man-in-the-middle for the two users. While serving as the MITM, the attacker can delay, change or destroy the data being sent.

The best countermeasures for integrity often come through appropriate authentication techniques. Replay attacks can be countered through an encrypted time sent value. The best countermeasure to message manipulation is an Message Authentication Code (MAC). While simple implementations of a MAC may do a sum of bits, MACs are often constructed with both cryptographic techniques and the authentication recognized between users. One common technique is to use a combination of Cipher Block Chaining and cryptographic hashes to construct a MAC [4]. This process ensures that each part of the message is included in the calculation of the MAC and the cryptographic key ensures only the desired party is able to create the MAC. The MITM attack can be countered through a combination of authentication measures, such as encrypted time values, MACs and public key authentication, which make sure that only the desired user is authenticated.

Availability within cybersecurity ensures that when data is needed it is possible to access it. Availability ensures that messages are able to be transmitted and received in a timely manner. Within VANETs, availability is important to maintain so that safety-critical messages can be sent and received in a timely manner. The most popular attacks on availability are Denial of Service (DoS) attacks. These attacks seek to overwhelm the receiver of the messages and prevent them from responding to real messages. Flooding is a subtype of DoS attacks which seeks to overwhelm the communication medium itself, and cause dropped packets due to the high amount of collisions between messages. Distributed Denial of Service (DDoS) is

a subtype of DoS attacks, which uses a number of compromised systems to create a DoS attack.

The primary defense for Availability is the identification and mitigation of DoS attacks. Identification of DoS attacks is typically accomplished through the use of Intrusion Detection Systems. Mitigation of DoS attacks can include disallowing attackers through lower level means, required authentication, revocation ability within that authentication, and honeypots. Disallowing attacks through lower level means could involve dropping packets or messages based on IP addresses or MAC addresses. Since this can be done earlier in message processing, less time is taken on each false message and can reduce the impact that a DoS attack can have. Required authentication can ensure that only authenticated individuals will have the ability to create DoS attacks on a system, which alongside the ability to revoke authentication, can theoretically create systems where DoS attacks are not possible, or are only possible once. Finally, honeypots create false areas which DoS attacks can be baited into targeting rather than the actual systems.

Authentication within cybersecurity ensures that a sender or receiver know the opposing receiver or sender. Authentication ensures that the identity being shown is the correct one. Within VANETs, authentication ensures that the vehicle who is sending a message is who they say they are, which can prevent false data attacks from compromised vehicles.

Attacks on authentication come in two forms, internal and external threats. External threats are theoretically easier to deal with, as once an authentication protocol has been put into place, so long as it does not have issues within it, external threats are ineffective against it. However, it is difficult to guarantee that authentication protocols are perfect, and even if they are perfect and cryptographically secure, that does not mean that they will continue to be secure in the future. Given the recent development of quantum computing, cryptographic algorithms that once would have taken hundreds of years to crack might become trivial [52].

Internal threats stem from the actions of a corrupted but authenticated user.

SCMS requires the usage of signature creation at bootstrap. This means that as vehicles are being created, their security keys and identity required by SCMS are installed before they even meet the road. As such, a majority of the focus of the security measures within SCMS are externally focused, as internal threats will need to have been authenticated at bootstrap. Internal threats often require the usage of misbehavior or intrusion detection systems. These systems will be covered in more detail in Section 2.2.3. External threats should be resolved through robust authentication processes, and by keeping up to date with upcoming authentication weaknesses.

Non-repudiation within cybersecurity is the means of holding people or entities responsible for what they transmit. Non-repudiation ensures that it is possible to prove that a malicious message that was sent by a user came from that user. Within VANETs, after misbehavior has been detected, the usage of the security credentials provided from SCMS at bootstrap ensure that messages which are signed by a vehicle are able to be traced back to that vehicle and revoked.

### 2.2.2 Security Credential Management System

Security Credential Management System (SCMS) is a Public Key Infrastructure (PKI) specifically designed for VANETs. However, just as importantly, SCMS is the current driver for changing and fulfilling abstractions of the security requirements which were created for the VANET standards. In this section, a system architecture of SCMS will be shown and the four main use cases of SCMS will be discussed.

The following system architecture for SCMS is taken directly from the IEEE report submission for SCMS. Throughout this section, the architecture will be explained and discussed.

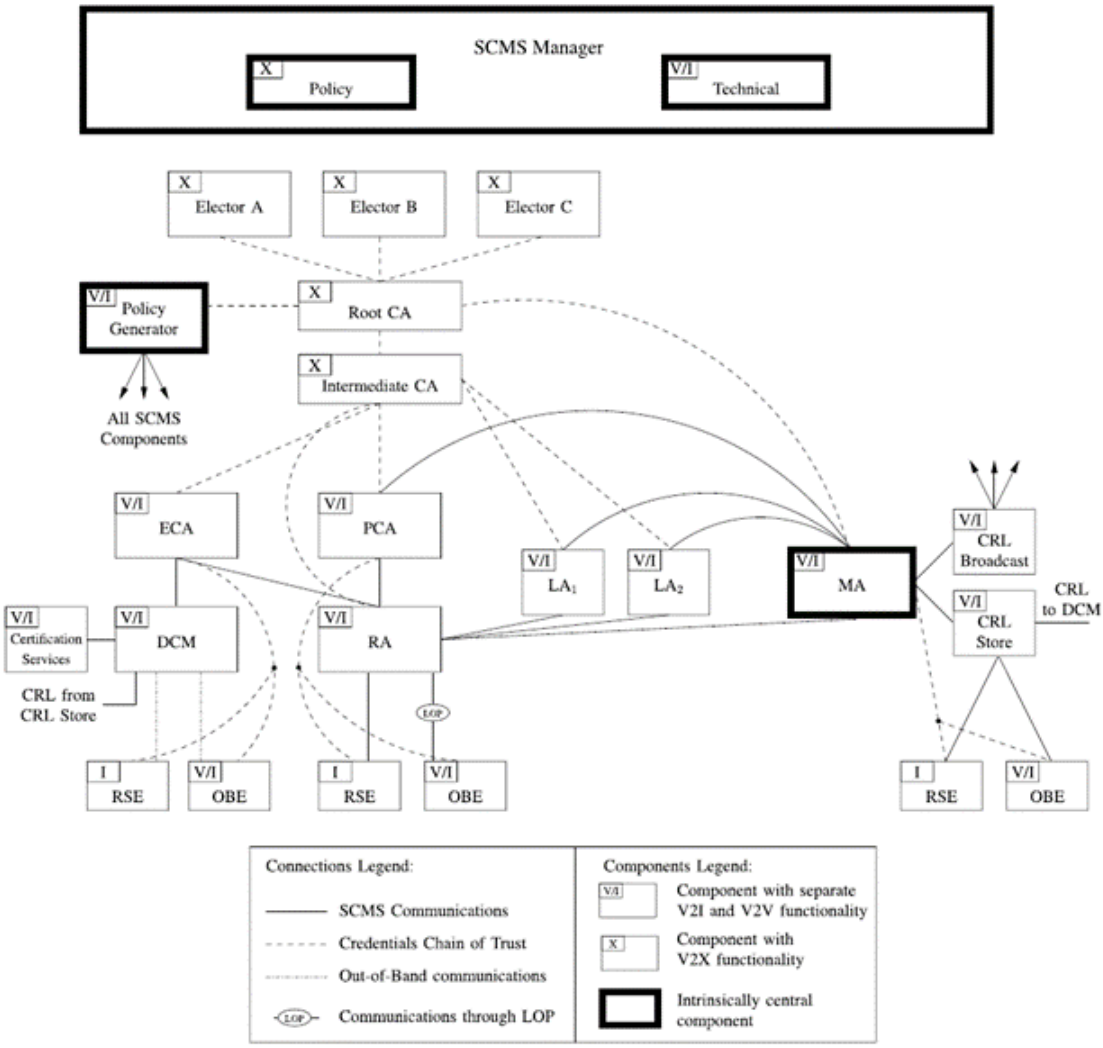



Figure 2.3: SCMS System Architecture Overview [10]

The SCMS manager creates policies which are efficient and fair to users, and defines the organizational goals and policies which are followed through the rest of the SCMS architecture. At the head of the Chain of Trust, electors are in charge of acknowledging or revoking the permissions of Root Certificate Authorities (CA). Root CAs are then responsible for managing and maintaining the lower system. They have final say in all decisions except those completed by the electors. The Root CA acts as the trust anchor which is present in other implementations of PKIs. The Policy Generator is responsible for creating and maintaining

the global configurations including the Global Certificate Chain File which contains all of the trust chains. The Intermediary CA acts as go between from the Root CA and the rest of the system to prevent high levels of traffic to it and to protect it from attacks.

The Enrollment CA (ECA) serves to give out enrollment certificates within the bootstrapping process. The Device Configuration Manager (DCM) guarantees to the ECA that the device undergoing bootstrapping is trustworthy. The Pseudonym CA (PCA) is one of the most important and most used parts of SCMS. Because of the need for privacy, pseudonym certificates are required, and must often be created in large quantities. The Registration Authority (RA) processes and forwards requests from devices towards the next CA or location that they need to travel. The RA is responsible for ensuring that revoked devices don't receive new certificates, and ensuring that attacks do not reach other isolated locations like the Misbehavior Authority (MA). The MA processes misbehavior reports to identify misbehaviors and malfunctions. It will pass the needed information to the Linkage Authorities (LA) to link pseudonym certificates back to their enrollment certificates to give back to the RA. The MA is also in charge of creating Certificate Revocation Lists.

The four main use cases for SCMS are defined as bootstrapping, certificate provisioning, misbehavior reporting, and certificate revocation. SCMS will authenticate vehicles being created in their bootstrap process through the ECA. The enrollment certificate is required for each vehicle to be able to interact and receive pseudonym certificates through the system. Certificate provisioning is completed through the distribution of pseudonym certificates which PCA generates. SCMS recognizes that some vehicles may not be able to communicate with the outside world for a significant period of time. Because of this, pseudonym certificates sets are distributed for a time period of up to three years. Within those three years, each pseudonym certificate is legal for 1 week, and there is a minimum of 20 active pseudonym certificates to ensure privacy requirements are met. Butterfly keys, specifically the butterfly

key expansion algorithm, is used to create these large numbers of certificates with specific time period legalities, which are able to hide the enrollment certificates from anyone but the LAs. Misbehavior reporting is completed by having the internal computers of vehicles, or On Board Units (OBU), submit misbehavior reports to the MA. The misbehavior reports contain information on the current vehicle status, misbehaving entity pseudonym certificate, and reason for submission. The behavior within MA's and the details within misbehavior reports has been abstracted for now. However SCMS has recently begun the implementation of parts of the misbehavior reports and the MA's abstraction [20].

Within this report, SCMS breaks misbehavior detection and correction up into two sections: Local Misbehavior Detection (LMBD) and Global Misbehavior Detection (GMBD). LMBD occurs within vehicles and is currently implemented only to include proximity plausibility content. The LMBD will create a report upon a found violation of the proximity plausibility and send it to the GMBD. The GMBD intakes all the misbehavior reports and will then evaluate and pass down judgement through use of a revocation or a warning. One crucial weakness within this system, which will be addressed within this project, is the lack of misbehavior detection outside of OBUs. RSUs or external IDS servers are not considered within the SCMS protocol.

### 2.2.3 Intrusion Detection Systems

Cybersecurity for any network can be divided into internal and external attackers. External attackers can be easier to prevent through cryptographic means, but internal attackers are assumed to have access to authentication for the system. For VANETs, internal attackers are assumed to have access to an enrollment certificate from SCMS as well as the pseudonym certificates that go with it. In order to protect against internal attackers, it is necessary for

a VANET to prevent attacks as well as be able to identify and respond to attackers. These are completed through the use of Intrusion Prevention Systems (IPS), Intrusion Detection Systems (IDS) and Intrusion Response Systems (IRS).

IPSs within VANETs have shown success in the studies completed researching them. [9] determined potential attackers within a system based on their trust scores earned through helping the system. [72] also found success in preventing sybil attacks through the use of their Priority Batch Verification Algorithm. [58] found success in the prevention of attacks by modeling the attackers' behavior using game theory.

IDSs seek to identify intrusion or misbehavior within a closed or secure system. IDSs are a popular research topic across all forms of networking, and their usage within VANETs is only a branch of their utilization in other areas. There are many different classifications of IDSs, and only a brief summary of them can be provided due to space requirements.

There are three IDS locations: Network IDS, Host IDS and Hybrid IDS. Network IDS (NIDS) act at higher levels within a network. They will capture network traffic and analyze it before sending it to the user. Host IDS (HIDS) typically act at the host level and will evaluate things like log files from the OS, and services and software executing on the host. Hybrid IDS combines NIDS and HIDS [68]. NIDSs are the most popular version within VANETs to evaluate network traffic coming from other vehicles.

IDSs will be structured as either Centralized or Distributed. Centralized IDS will have multiple agents analyze an issue and then report to Central Command and Control (C&C), which will then decide upon a course of action. The LMBD and GMBD suggested by SCMS falls within this category. Distributed IDSs will either be a Collaborative Distributed IDS (CDIDS) which is a system of agents that work together to collect data and make decisions collectively, or a single standalone IDS. [53] utilizes a CDIDS system through the use of

blockchain-assisted detection.

IDS detection techniques will fall into one of three categories: Anomaly-Based, Knowledge-Based, or Hybrid. Within Anomaly-Based IDSs, a model will seek to learn a profile or heuristic for normal behavior and then detect when there is a deviation from normal behavior. This has the benefit of being able to detect zero-day attacks since they will be outside the normal behavior of the system but will also have a number of false positive results. Zero-day attacks are cyberattacks who are not currently known and are being launched at a system for the first time ever. Knowledge-Based IDSs, on the other hand, match attack patterns and signatures to those already found in a database. Because of this, they are able to have very low false positive rates, but also are unable to detect zero-day attacks. Hybrid detection techniques will combine both in order to increase accuracy in identifying attacks. All three techniques are common within VANET research. [62]'s usage of plausibility checks falls within Knowledge-Based techniques, while the usage of Long Short Term Memory (LSTM) and Convolutional Neural Networks (CNN) that [3] employs falls under Anomaly-Based techniques.

Of the Knowledge-Based Techniques employed in VANETs, Rule-Based reasoning and Case-Based reasoning are the most common. Rule-Based reasoning is uses a series of if-then logic chains which are set by humans manually. This is often the same as writing code which will check for certain requirements. Case-Based reasoning will choose the best outcome for a situation based upon similarities to past situations. However, this requires a history log to be kept of past situations which may not be possible for the resource-constrained OBUs.

Metrics within IDSs are some of the most important within any field, as failing to detect or rectify a false positive can be extremely time consuming and be a potentially dangerous action. Upon failure to detect a false positive, the system can be impacted, which within the scope of VANETs could lead to slowdowns or even loss of life. For vehicles operating

within a fully autonomous system, a false positive could potentially leave a vehicle nearly unusable until the mistake has been rectified. The following metrics are commonly used for evaluating IDSs:

- Detection Rate (TP/DR) – Percentage an IDS identifies an attack when there is one
- False Positive (FP) – Percentage an IDS identifies an attack when there isn't one
- False Negative (FN) – Percentage an IDS doesn't identify an attack when one occurs
- Accuracy – Percentage an IDS identifies correctly
- Precision – Percentage there is an attack when an IDS identifies an attack
- F-Score – Harmonic mean of precision and DR
- Recall – The ability for an IDS to identify a misbehaving node
- Evaluation Time (ET) - The time that it takes for a IDS to analyze a data point
- Detection Time (DT) - The amount of time that it takes for any node to identify an attack

When identifying or classifying the data which will be used to train an IDS, the anomaly type is important to consider as well. Anomalies can be classified in one of three ways: Individual/Point Anomalies, Contextual Anomalies, or Collective Anomalies. Point Anomalies are data points which are anomalous regardless of context around them. Contextual Anomalies are those which are anomalous within their context and Collective Anomalies are collections of items which are anomalous when compared to other collections [16]. These anomaly types are of critical importance for VANETs to be able to decipher between for detection. In order to best detect these different types of anomalies, the number of data points contained within each sequence to be analyzed by a IDS is of great importance. [67] conducted sensitivity testing on sequence length within VANETs when training LSTM and other Sequence-Based models.

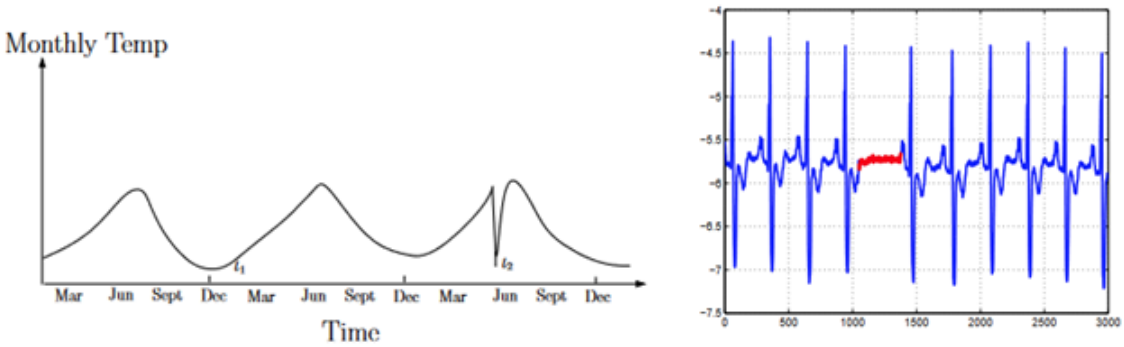



Figure 2.4: Contextual and Collective Anomalies [16]

The procedure for passing data through a IDS is preprocessing, modeling and postprocessing. Data preprocessing ensures that the data being passed into the model is appropriately constrained before being analyzed. Preprocessing may include data formatting, removing illegal values, discretizing data, removing outliers, normalizing data, and feature selection [34]. Removing outliers and normalizing data ensures that abnormal data is removed before being processed. This ensures that a sensor malfunction is not evaluated as a correct piece of data. In an image, a single pixel which contains maximum color values but is not consistent with the nearby pixels could be taken as an outlier and then be normalized to what typical data would be expected for that area.

Feature selection is an incredibly important topic for model creation and use. Because of the massive amount of data that is being sent and processed, only taking the parts of that data that are necessary for correct outputs is crucial. While it would be best to evaluate a 4K image at a rate of 60hz, this amount of data is unlikely to be necessary for object identification and evaluation. Before deployment of models, feature selection techniques are often run to determine the needed features for a model to be able to produce optimal outputs. Feature selection within VANETs is incredibly important as not only do they have the normal features of network traffic, but they also have kinematic values and vehicle system

data which need evaluating. [37] found over 150 possible features within VANET systems and evaluated 95 of the ones most likely to be relevant.

In order to extract the results needed from the preprocessed data, each data point is passed into trained models. These models are trained through machine learning, optimization, or other means. There are huge numbers of models, training methods, and theories which are used for determining actions or classifications to be taken from observations, and a comprehensive study and evaluation of all of them is not possible. While remaining concise, the following information will look to discuss the definitions and categorization of different models and training methods analyzed in this thesis, many of which are based upon the information available in [68].

One incredibly important aspect of training models is the availability of training labels. Training labels are the correct output labels for the corresponding input data. In general, if training labels are available, the model can be trained to have more accurate results. However, training labels are not always available. Simulated data sets will often have high availability of training labels, but may not be accurate to real world scenarios. Due to the availability of training labels, only certain training methods may be possible.

Supervised training is conducted upon models which use fully labeled data. These training methods will often create extremely accurate or precise models so long as the problem itself is not overly complex. However, supervised training can be impacted by skewed training data and has the potential to overfit to training data. Overfitting can be reduced through the use of K-Fold cross validation, which splits training and testing data. If the model performs well on the training data, but poorly on the testing data then it is likely overfitted. Unsupervised learning is training which occurs without training labels. This is done by assuming that within the data set normal behavior is the most common and therefore, anomalies are abnormal behaviors.

Within IDSs the most common return values from a model are classification data, either binary or multi-class, which identify the presence of or type of intrusion [14]. Postprocessing may convert the data into a machine or human readable format or enable the results to be used in future evaluations of the IDS.

A variety of model classifications and types are available for different supervision levels and availability of training labels. Classification and Reinforcement Learning are the two most popular model types within VANETs.

Classification Techniques are supervised learning models which output categorical data. The input data is run through a predictive function with an initial set of weights or parameters and the output classes are recorded. The difference between the expected and actual output classes from the predictive function is defined as the loss function, which the model then attempts to minimize. The main techniques used for VANETS within this classification are Support Vector Machines (SVM), K-Nearest Neighbors (KNN), Decision Trees (DT), Random Forest (RF), Naïve bayes classifier (NB), Artificial Neural Networks (ANN), and Long Short Term Memory (LSTM). SVM, KNN, DT and RF will described in more detail in Chapter 3.

Naïve Bayes Classification is based upon the Bayesian theorem, which predicts the probability of output classes based upon the probability of attributes falling into those classes [78]. The Naïve Bayes Classifier assumes independence among input attributes, which means that feature selection is extremely important so as to not double count connected input attributes.

Artificial Neural Networks (ANN) consist of connected artificial neurons which evaluate the data as it flows in one direction through the connections between neurons. ANNs are typically split into three internal layers: input , hidden, and output layers. The weights and biases along the connections and neurons are adjusted through the learning process. In

order to avoid the exploding gradient problem, where weights of one layer will continuously increase while those of the next layer will continuously decrease, ANNs are trained through the use of backpropagation.

Long Short Term Memory (LSTM) is a Recurrent Neural Network, meaning that the data that passes through a ANN does not only flow in a singular direction [77]. LSTM are best used when evaluating time series data, such as a sequence of BSMs. LSTMs remember a cell state while evaluating a series of data points, and modify that state through the use of a input gate and a forget gate. A part of the cell state is then passed to the output gate to be combined with the next data point before being evaluated by the next input gate.

Reinforcement Learning is a model type which learns to maximize performance measures within an continuous environment. Reinforcement Learning algorithms will typically collect samples of mappings of actions to environmental changes and then use value function approximators to evaluate these mappings. Two of the most common versions of Reinforcement Learning Algorithms are Markov Decision Processes and Q-Learning. A Markov Decision Process contains a set of states, possible actions, conditional probabilities for translating from a state and action to the next state, and a reward or cost function based on the taken action. Q-Learning is an ANN technique which learns a Q function to approximate the reward function.

### 2.2.4 Current IDS Techniques in VANETs

Intrusion Detection Systems and Misbehavior Detection Systems are extremely similar, with the primary difference being that that misbehavior detection will be sent to the MA and lead to a revocation of a certificate, while an intrusion detection will be sent to C&C for further evaluation for necessary action. Within the realm of VANETs and SCMS, we will

consider the two detection systems to be the same, and the procedures used in both to be interchangeable.

Until recently, no data sets were available within VANETs, and as a result, it was necessary to use self-created data sets for evaluation. Recently, three standardized data sets have started seeing use in VANET research: VeReMi, Goncalves, and the ToN-IoT. All but ToN-IoT are currently synthetic data sets, which are created through simulations.

Self-created data sets are those which the researcher programs attacks and defenses specifically for their own paper. Self-created sets may seem to have an uncertain ability to correctly evaluate techniques due to the likelihood of producing a biased data set which would shed a positive light on their technique. However, the purpose of many of the papers is to demonstrate proof of concept rather than a final model, which a self-created data set is able to do.

The Goncalves data set was created using SUMO as the traffic simulator and Network Simulator 3 (ns-3) as the network simulator [26]. The data set hosts DoS attacks and fabrication attacks. The data set also does not use a traditional BSM message, but rather a Context Awareness Message (CAM). The CAMs sending rate is dependent not on a static value, but is dynamically influenced by factors such as acceleration, position, speed and heading. They claim that uniform sending times will be unlikely to be used in the real world applications of VANETs, and that data sets which use a uniform sending time for messages allow for easier DoS identification than what would be possible. The Goncalves set then published a second paper evaluating their data set, the feature selection techniques used for their data set, and the difference in results when training upon a single data set or upon a combination of data sets.

The ToN-IoT data set is the only real world data set, but draws data from Internet of Things

network traffic rather than VANET traffic [24]. It has also only been used in a single paper, however that paper evaluated a total of 8 different models on the data set, as well as a number of feature and parameter selection choices.

The final and most popular data set is the VeReMi and VeReMi Extension data sets [33, 70]. The VeReMi data set is a simulated data set and is based upon the communication model between an RSU and an OBU. To generate their data set, they make use of the Framework for Misbehavior Detection (F2MD) [32]. They utilize VEINS and the Luxembourg SUMO Traffic (LuST) scenario [19]. The LuST scenario is a traffic simulation scenario validated with real driving data. VeReMi utilizes a subsection of the LuST scenario within their simulation. The VeReMi data set contains only four attacks, while the VeReMi Extension data set additionally adds three malfunctions and two more attacks which are combined or expanded for a total of 19 scenarios.

Malfunctions of position and speed impact the values within each BSM by containing either constant values, constant offsets, random values or random offsets. The final malfunction is Delayed Messages, which will adjust the sending time of a BSM within a margin of error that might occur within a real OBU.

The Disruptive attack is a replay attack using messages from different senders. The Data Replay attack resends messages from a specific node. The Eventual Stop attack creates a fake vehicle ahead of it using a pseudonym and then instructs the fake vehicle to pretend to stop, enabling the attacking vehicle to slow to a stop. The DoS attack sends more frequent real BSMs, while the DoS Random attack sends frequent nonsense data. The DoS Disruptive attack acts as a flooding attack targeting the network itself. The Data Replay Sybil replays data from a target node, all the while changing the pseudonym for each message sent. The DoS Random Sybil sends out random data at a rapid rate using different pseudonyms. The DoS Disruptive Sybil acts as a flooding attack on the network, sending messages under the

maximum number of pseudonyms that a vehicle has access to. Finally, the Traffic Congestion Sybil is a large-scale Eventual Stop attack, which creates a number of false vehicles using pseudonyms, and attempts to stop the entire traffic flow.

[13] explores the usage of a SVM on a Vehicle-Based IDS to detect false-data attacks within ITS. This paper observed an attack detection time of 0.11 to 0.13 seconds for sybil attacks. [51] also explores the detection of sybil attacks. This paper uses an Extreme Learning Machine (ELM) to locate entropy changes in message and data rates to identify sybil attacks. The paper was able to achieve nearly 100% accuracy after training for 500 epochs.

[53] explores the usage of a blockchain as a CDIDS to detect various misbehaviors. Their model, BLAME, was able to achieve an accuracy of 100%, 78.5% and 49.7% for false data injection, packet dropping and replay attacks respectively. Reputation and trust schemes are common models used within VANET IDSs. [50] uses a trust evaluation system to identify DDoS attacks. [42] and [39] both use trust systems to identify misbehaviors. [38] uses a Markov reputation scheme to increase the detection rate, decrease the detection time, and decrease overhead to identify attacks compared to other game theory alternatives.

The next set of models which will be discussed are those evaluated on the original VeReMi data set. A review of papers using the VeReMi Extension data set can be found in Chapter 3. [29] utilizes machine learning options such as K-Nearest Neighbor and Decision Trees to evaluate the VeReMi data set, maintaining a precision of over 94%. [63] introduces Received Signal Strength Indicator as a new feature which evaluates the distance from the receiving car to the sender, which can help identify when the true value and the stated distance in the BSM do not match. The final paper, [64] integrates Rule-Based reasoning to create misbehavior reports, which are then passed into a K-Nearest Neighbor and Support Vector Machine for analysis. All of this analysis occurs within a Vehicle-Based IDS system.

### 2.2.5 Intrusion Response Systems

The final aspect within security requirements, IRSs, have had minimal research conducted on them. To the best of my knowledge, IRSs are treated in two ways within current literature. The first method of treatment is as a pass-through for the results of a IDS, where if there is a detection, then the certificate is revoked. [10] goes into detail on the process which SCMS uses in the revocation of a certificate which includes the expansion of the butterfly key to obtain the enrollment certificate as well as the sending of the Certificate Revocation List.

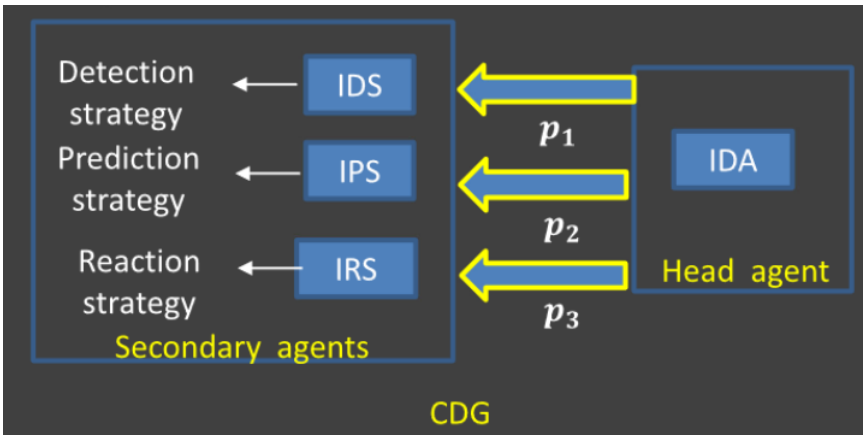



Figure 2.5: Intrusion Decision Hierarchy [59]

The second method of treatment for IRSs is one in which the IRS is able to make certain decisions based on the results of the IDS. [59] proposes a hierarchical game theory implementation. Within this implementation the IPS, IDS and IRS are under the control of a Intrusion Decision Agent (IDA), which is the major actor in a cyberdefense game against infected vehicles. Within this game, the IRS has three possible actions it can take: revoke a certificate, change its own pseudonym to prevent tracking, and periodically update its cryptographic keys to ensure data privacy. All three actions are reactions to different attacks which an attacker may perform.

Within all literature, there are substantial areas which are missing significant analysis or

testing. The first concept this can be seen in is detecting an attack and taking an action, which is seen as neutralizing an attacker in existing literature. While this may be true for other network types, VANETs are cyber-physical systems. As such, there is a need to question how the physical aspects of the system will remain impacted with a corrupted vehicle remaining on the road. Along the same lines, while a false positive result may be annoying for a computer user, there are potentially lethal consequences for false positive results within VANETs. If an OBU marks a blind spot notification as untrue, it may attempt to merge lanes straight into another vehicle. Therefore, even if an assumption is made that an attacker will not try to cause any harm through the use of its vehicle's physical body, the risk of negative impacts from decisions made based off of IDS results remains extremely high.

# Chapter 3

# RSU-Based Intrusion Detection

## 3.1 Abstract

*Abstract*- Vehicle Ad-hoc Networks (VANET) are of critical importance to secure and protect from malicious actors. Transmission of inaccurate or incorrect information can inflict serious dangers within the cyber-physical VANET networks. Intrusion Detection Systems (IDS) identify misbehaving nodes while balancing efficiency and detection rate. Current research is focused on maximizing detection within VANETs, but fails to consider the structure of the system. This paper introduces an RSU-Based Intrusion Detection System (IDS) which utilizes the current proposed VANET structure. The IDS provides location specialization to increase accuracy while increasing privacy, reducing system complexity and costs, and reducing data flow compared to other state of the art alternatives. The proposed IDS system is simulated, trained and evaluated on several Machine Learning alternatives. The proposed method is found to be able to produce an average accuracy of 99.75% and a average precision of 99.28% across its best models.

## 3.2 Introduction

VANETs are utilized within Intelligent Transportation Systems (ITS) to provide safer and more efficient movement. To accomplish this, vehicles and Roadside Units (RSU), will

regularly communicate to update other nearby participants with data that they need to drive safely and efficiently. Cyber-physical systems require higher level of safety in order to protect both the security of the On-Board Unit (OBU) as well as the health of the passengers.

OBUs communicate with other vehicle's OBUs and with RSUs while using the Security Credential Management System (SCMS) to prevent external access and communication within VANETs [10]. Internal access to SCMS is granted at bootstrap, or on the creation of devices. However, after bootstrap, it is impossible to grantee that the OBUs will remain untouched and in perfect working order. Therefor, IDSs are utilized to detect misbehaving or faulty OBUs on the interior of the SCMS security protocols.

The most common form of data transmission with VANETs will be the Basic Safety Message (BSM). A BSM will contain standardized data about the vehicle sending it, such as position, speed, acceleration and heading [47]. Each OBU will send a BSM at least once every 100ms so long as the network is not congested. Common attacks will adjust the sending rate of BSMs, replay previously sent BSMs, or modify the data being sent within a BSM. The required usage of pseudonyms within the SCMS protocol for privacy concerns has also has severe cybersecurity implications. While pseudonyms do fulfil the requirement of privacy, the pseudonyms can only be linked back to an original vehicle after reports against the pseudonym are made [10]. This means that a corrupted vehicle has a minimum of 20 pseudonyms to conduct attacks with before they are reported and the pseudonyms are able to be linked back to the vehicle and revoked.

SCMS contains four main use cases: bootstrapping, providing certificates, misbehavior reporting and revoking certificates. This paper focuses on misbehavior reporting, which utilizes the following SCMS components: OBUs, Registration Authority (RA), and the Misbehavior Authority (MA). Within SCMS, the proposed misbehavior reporting protocol contains Local Misbehavior Detection (LMBD) and Global Misbehavior Detection (GMBD) levels. Misbe-

havior reports are generated at the LMBD level by OBUs and then routed to the MA to make final decisions at the GMBD level [20]. The RA serves as an intermediary between them and removes identifying information from reports and ensures that the MA does not receive malicious messages. RSUs may or may not be abstracted away as a method to forward the reports from the OBUs to the RA.

As with much research on Internet of Things, OBUs are assumed to resource contained devices. This means that the OBUs will have limited computing power, either due to low specifications or due to needing to maintain numerous applications. Thus, it is assumed that OBUs will be able to perform basic plausibility checks for detection, but may be unable to run heavier machine learning or deep learning models.

To avoid the resource constrained devices, state-of-the-art research introduces RSUs and an IDS hosted on a separate server as a potential solution. The RSUs are used to forward all BSMs sent within a system, and the IDS server is used for misbehavior detection utilizing more information than would be available for a single vehicle. The IDS server is assumed to not be resource constrained, and is able to run more complex detection processes, such as deep learning, deep reinforcement learning, and machine learning.

This paper looks to demonstrate the effectiveness of an IDS placed within an RSUs as compared to externally hosted IDS systems. The RSUs are able to analyze locally received BSMs at a higher level of accuracy and efficiency due to their ability to specialize in a specific location. Because they serve an an intermediary to the RA, they are also able to incorporate the reports already sent by vehicles into their analysis. Finally, by eliminating the need for a dedicated IDS server, this system can improve privacy, reduce system complexity and costs, and reduce data transmission.

## 3.3 Literature Review

Recent research of IDSs in VANETs has been focused on three distinct areas, the creation and solution to specific attacks, efficient detection methods for the resource constrained OBUs, and IDS methodologies within the IDS server configuration. As real world data is not currently available, simulation data has been created and used for the testing of IDSs. The Framework for Misbehavior Detection (F2MD) is a simulation framework built upon VEINS, which simulates various attacks and communications which will occur within a real VANET system [32]. The VeReMi Extension data set, was constructed using the F2MD Framework and contains a total of 19 attacks[33]. The VeReMi data set contains a ground truth file, as well as a log of data from each vehicle including their true pathing as well as the BSMs they receive from other vehicles.

The authors of [56] analyze the sudden-stop attack within the VeReMi Extension data set and train a reinforcement learning model for detection. [67] utilizes a two step prediction model, first detecting any misbehavior and then classifying each misbehaviors. The authors only utilize position and velocity, and utilize chains of messages, ranging from 5 to 30 messages. Deep Belief Networks, Long Short Term Memory, Gated Recurrent Units, and Random Forest are compared within this work. [62] utilizes plausibility checks and ensemble machine learning, but only evaluate 5 of the 19 attacks.

The authors of [3] use Multi-Edge Access Computing, and consider the DoS, Sybil, Data Replay and Disruptive attacks. They use Deep Learning Engines (DLE), and consider the conversion of vehicle data into images to better utilize the DLEs. [2] utilizes Convoluted Neural Networks, Long Short Term Memory, and Gated Recurrent Networks which are trained on a cloud server for detection. [22] utilizes Random Forest and K-Nearest Neighbors along with new features, such as Received Signal Strength Indicator, to identify the constant

offset attack.

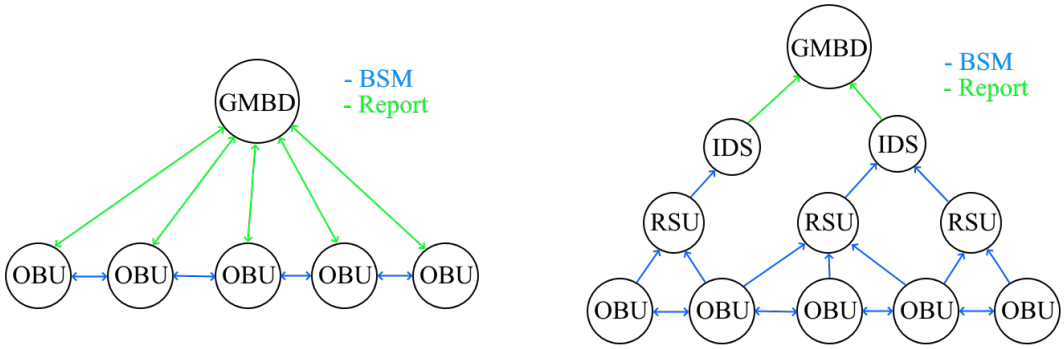
Several previous papers [2, 61] claim to run identification on RSUs, but utilize the full data set and make assumptions that every message sent is received and forwarded perfectly to each RSU that needs it. This is unrealistic behavior for RSUs, and should be classified as cloud computing instead. To the best of our knowledge, this is the first paper which utilizes the F2MD environment or a standardized data set to create real RSU units which train and utilize messages which would actually be received by a real RSU.

## 3.4 Methodology

### 3.4.1 Network Scenario

As stated before, this project is the first to implement and train an RSU-Based IDS in a standardized data set or framework. The data utilized within the training was not from the VeReMi Extension data set, but instead was generated from the F2MD framework which created that data set. The traffic network uses the Luxembourg SUMO Traffic (LuST) scenario, which contains real world backed vehicle data [19]. Each vehicle uses its OBU to send periodic BSMs which may contain misbehaviors. Previous network topology has been implemented in one of two ways shown in Figure 3.1.

These topologies contain three unique node types: GMBD, LMBD and consolidation nodes. Consolidation nodes intake reports or BSMs from several lower level nodes, and forward them up the hierarchy. A LMBD node will evaluate BSMs and generate reports and the GMBD will evaluate reports to determine the action to be taken. Within the second image in Figure 3.1, The RSUs are used as consolidation nodes, and the IDSs are used as LMBD nodes.




(a) Traditional GMBD Network (b) Traditional IDS Network

Figure 3.1: Common VANET Network Topologies

We propose a combined LMBD/consolidation node as shown in Figure 3.2. This node will intake BSMs and reports, and then perform LMBD upon both the BSMs and the reports. This combination node is able to use additional information included within the reports and specialize in a single location. As the input complexity for a problem increases, the accuracy of a solution has been shown to decrease [60]. Since each RSU specializes in the geographic area it can receive messages from, the RSU should show improved accuracy compared solutions which consider larger input areas.

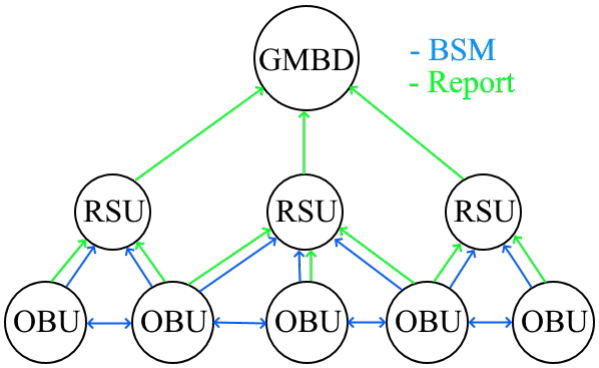



Figure 3.2: Proposed Network Topology Including a Hybrid Node

Our modification to F2MD includes the placement of RSUs at 21 of the most popular intersections within the modified LuST scenario. Vehicles were also modified to send out their misbehavior reports through beacons, the same as the BSMs.


Figure 3.3: Placement of RSU units within the F2MD Framework [33]

The VeReMi Extension parameters were then run through the modified F2MD network, with each RSU creating a new data set from the BSM and report messages received. The VeReMi Extension data set contains a total of 19 attacks, which are all conducted again within the modified F2MD network. The attacks are: (1) Constant Position, (2) Constant Position Offset, (3) Random Position, (4) Random Position Offset, (5) Constant Speed, (6) Constant Speed Offset, (7) Random Speed, (8) Random Speed Offset, (9) Delayed Messages, (10) Disruptive, (11) Data Replay, (12) Eventual Stop, (13) DoS, (14) DoS Random, (15) DoS Disruptive, (16) Data Replay Sybil, (17) Traffic Congestion Sybil, (18) DoS Random Sybil, (19) DoS Disruptive Sybil.

### 3.4.2 Data Processing

The newly constructed data set, much like the original VeReMi data set, is compromised of two file types, a ground truth file and individual data files for each RSU. The ground truth file contains the data shown in Figure 3.1. This ground truth file contains one significant change from the original data set, that being the inclusion of a misbehavior field. Other studies either trained the models in a unsupervised setting, training only on non-misbehavior data, or have determined some other way to determine misbehaving vehicles. Plausibility checks, difference from typical behavior, or other calculations have been used to compare the messages being sent and recieved to the true condition of the vehicles stated in their separate vehicle data files. However, these solutions have issues with the validity associated with their detection. The usage of plausibility checks or calculations ensures that rather than teaching the IDS how to detect the attacks, the models are instead trained to detect the calculations used to define the misbehavior.

| Field Type | Contains | Data Type |
|---|---|---|
| **Message ID** | - | Integer |
| **Sender ID** | - | Integer |
| **Sent Time** | - | Float |
| **Received Time** | - | Float |
| **Position** | X, Y, Z | Float |
| **Position Error** | X, Y, Z | Float |
| **Speed** | X, Y, Z | Float |
| **Speed Error** | X, Y, Z | Float |
| **Acceleration** | X, Y, Z | Float |
| **Acceleration Error** | X, Y, Z | Float |
| **Heading** | X, Y, Z | Float |
| **Heading Error** | X, Y, Z | Float |

Table 3.1: Ground Truth Data Types

Our solution was to modify the F2MD framework to instead output a misbehavior value if the vehicle is misbehaving. This ensures that the misbehaviors that we are training our models

on are actual misbehaviors and not a calculation being completed. While this solution does provide more accurate data for when any vehicle believes it is misbehaving, there were two issues which were discovered and are covered in more detail in Section 3.5.1.

The second file type created for this project was the messages recieved by each RSU. This file type again functions similar to the original VeReMi data set, with one significant change, the inclusion of misbehavior reports generated by vehicles. Each report was formatted as shown in 3.2, containing data from the reporter and the reported, as well as information on the plausibility checks conducted by the vehicle.

| **Field Type** | **Contains** | **Data Type** |
|---|---|---|
| **Reporter ID** | - | Integer |
| **Reported ID** | - | Integer |
| **Report Sent Time** | - | Float |
| **Reported Message Sent Time** | - | Float |
| **Plausibility** | Proximity, Range, Position, Speed | Float (-20.0, 1.0) |
| **Consistency** | Proximity, Speed, Position/Speed, Max Position/Speed, Position/Heading | Float (-20.0, 1.0) |
| **Kalman Filter** | PACS, PCC, PSCP, PSCS, PSCSP, PSCSS, SCC | Float (-20.0, 1.0) |
| **Other** | Sudden Appearance, Beacon Frequency | Float (-20.0, 1.0) |

Table 3.2: Report Data Types

The number of previous messages used within data sets is an important determination to make. [2, 3] both use a sequence length of 20 messages while [67] evaluates sequence lengths from 5 to 30 messages. Sequence length is one of the primary draws to a Server-Based IDS, as individual RSUs are only able to utilize the data recieved while the vehicle is within its range. However, since F2MD and VeReMi only send out BSMs every second, it is unrealistic for an RSU to have access to data occurring over a period of 20-30 seconds. In order to

demonstrate the power of location specialization, a sequence length of 2 was utilized within our implementation. In addition, messages containing pseudonyms with no prior messages were still evaluated.

| Field Type | Contains | Data Type |
|---|---|---|
| **Position** | X, Y | Float |
| **Speed** | X, Y | Float |
| **Acceleration** | X, Y | Float |
| **Heading** | X, Y | Float |
| **Received Time** | - | Float |
| **Number of Reports** | - | Integer |
| **Plausability Checks** | See Table 3.2 | Float (-20.0, 1.0) |
| **Consistency Checks** | See Table 3.2 | Float (-20.0, 1.0) |
| **Kalman Filter Checks** | See Table 3.2 | Float (-20.0, 1.0) |
| **Other Checks** | See Table 3.2 | Float (-20.0, 1.0) |
| **Position Change** | X, Y | Float |
| **Speed Change** | X, Y | Float |
| **Acceleration Change** | X, Y | Float |
| **Heading Change** | X, Y | Float |
| **Time Between Beacons** | - | Float |

Table 3.3: Training Data Types

The two data files were then processed to create trainable data. First, the recieved message file was split into two separate files for parsed BSMs and reports. The BSM file was then used as the base to construct the training data. Each BSM was checked against the ground truth file to determine if it was caused by a misbehavior, checked against the report files to determine if a report had been generated for the BSM, and finally checked against the pseudonym IDs within the BSM file to determine if there was a previous message sent by that pseudonym to generate a sequence. This process created a data set with a total of 36 features, which are shown in 3.3.

Finally, traditional machine learning preprocessing techniques were applied to the data. Each feature removes the mean and scaled the values by unit variance to prevent biased estimation

within modeling. This places each value centered on zero, scaled according to the standard deviation of the feature.

### 3.4.3 Models

As the purpose of this paper is to determine the potential usage of an RSU-Based IDS system, traditional models used within VANET IDSs were used for a fair comparison. This paper utilizes Support Vector Machines, K-Nearest Neighbors, Decision Trees, Random Forest, and a voting ensemble of the other four classifiers. A Support Vector Machine (SVM) is a method which locates the best hyperplane to maximize margin size through high or low dimensional space [45]. SVMs will also incorporate slack variables, $\xi$, which allow the breaking of margins for outlier data points. K-Nearest Neighbors is an ensemble voting method, which utilizes the classification of nearby, or similar, data points to itself [49]. K-Nearest Neighbors requires normalization of data to prevent overfitting to certain features. A larger K values will reduce the impact of noise on a classification of a single item but make the boundaries between classes noisier.

Decision Trees create a set of rules which classify the input data based upon splitting conditions. Because the decision trees are binary splits, the training time for decision trees are logarithmic, which is significantly lower than most models. However, because of how low the cost to train is, it's important to set a max depth and prune the tree leaves to prevent overfitting the input data into their output classifiers. One such solution to overfitting Decision Trees is the usage of Random Forests as an ensemble method [8]. Within Random Forests, a number of decision trees are created and vote upon a result. Majority voting is then used to determine the classification of a data point.

The usage of the voting ensemble method is shown in Figure 3.4. In this method, each of

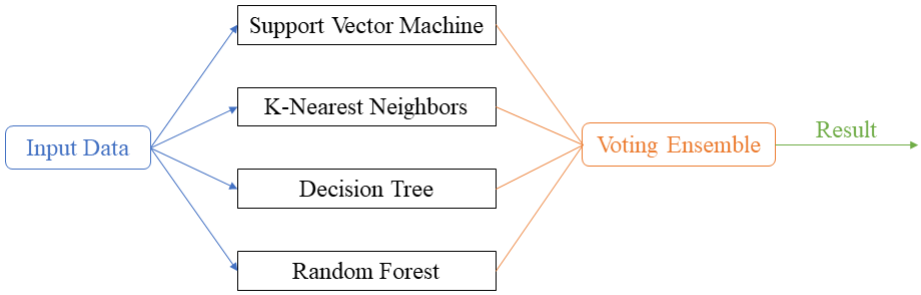



Figure 3.4: Voting Ensemble Structure

the four models will individually evaluate the input data. A majority voting of the results is then conducted. If a tie occurs within the ensemble voting, the vote from RF was used as the tiebreaker since its performance was able to maintain the highest accuracy on average. This method is able to perform both better and worse than the rest of the models because of the voting system. However, because of the voting, it is able to remain resilient to any outliers which are present in the individual models.

The sklearn python library was used to train and test the models [48]. K-Fold cross validation of the models was completed. The data from each RSU was broken up into two sections, with 75% used to train the models and 25% used to test the models.

Each model was tuned through the use of hyper-parameters and feature selection for each attack. Evaluation of hyper-parameters was conducted through the use of a gridsearch, effectively evaluating every possible combination of hyper-parameters. The different hyper-parameters considered for each model, and possible values, are included in Table 3.4. Each hyper-parameter used 5 fold validation to evaluate each hyper-parameter combination 5 times, and determine the best combination according to the average accuracy throughout the cross validation.

Finally, feature selection was performed for each attack. In order to evaluate the features to be used, a recursive feature elimination was completed. An evaluation was performed on all

| Model Type | Hyper-Parameters | Values |
|---|---|---|
| **Support Vector Machine** | C (Regularization Parameter) | 1, 10, 100 |
| | Kernal | Linear, RBF |
| **K-Nearest Neighbor** | K (Number of Neighbors) | 3, 5, 7 |
| **Decision Tree** | Max Depth | 50, 100, 500 |
| **Random Forest** | Number of Trees | 100, 500, 1000 |
| | Max Depth | 50, 100 |

Table 3.4: Hyper-Parameters Considered in Grid Search

features and the the least important feature was then removed. This process repeats itself until there is only one feature left. This process used 5 fold validation to ensure that the feature being removed was accurate. After a list of features for each number of features has been generated, the models were trained on each possible list of features. The model with the highest performance among these was then selected as the final model and the final list of features to be used. Figure 3.5 shows the accuracy results for Random Forest throughout this process for the Replay Attack.

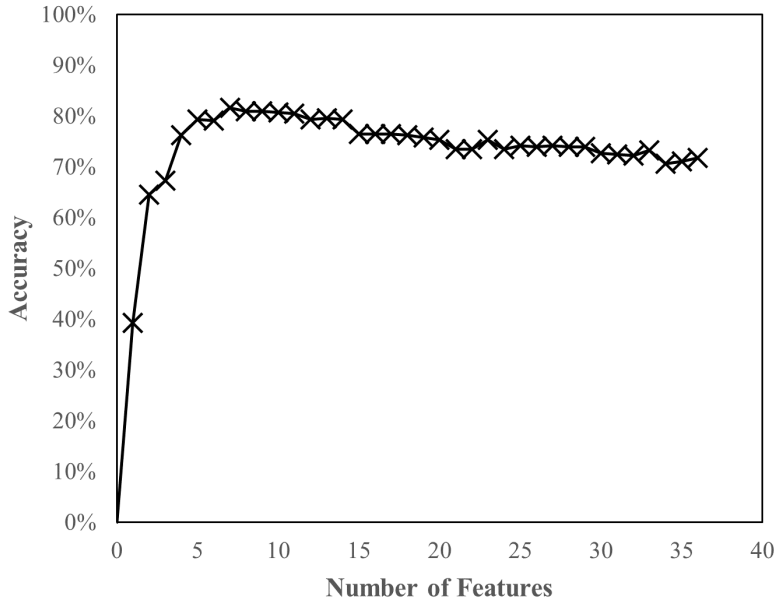



Figure 3.5: Random Forest Feature Selection Accuracy for the Replay Attack

Since this conducted for each attack type, different features may be selected based on the attack parameters. For example, the DoS attack relies heavily on time since last message, while the random position attack may not need the time since last message at all. Two examples of features selected for the Random Forest Model are included in Table 3.5.

| **Attack Type** | **Number of Parameters** | **Features** |
|---|---|---|
| **Replay Attack** | 7 | Position (X, Y),<br>Position Change (X, Y),<br>Speed Change (X, Y),<br>Heading Change (X) |
| **DoS Attack** | 9 | Max Pos/Vel Consistency,<br>Beacon Frequency,<br>Position Change (X, Y),<br>Speed Change (X, Y),<br>Acceleration Change (X, Y),<br>Time Since Last Beacon |

Table 3.5: Random Forest Feature Selection For Replay and DoS Attacks

## 3.5 Results

The modified F2MD was run for each attack and the data was then processed. The data was split, and the models were trained on 75% of the data to maximize accuracy. The remaining 25% of test data was then evaluated. When evaluating IDS behavior within the VeReMi data sets Accuracy, Precision, Recall and F1-Score are the most commonly used metrics. These metrics are based upon True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). True Positive and True Negative indicate when the model correctly identifies a classification, while False Positive and False Negative indicate the model incorrectly identifying a classification.

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

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

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - Score = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

The results across all attack types are located in Table 3.6, Table 3.7, Table 3.8, and Table 3.9 for Accuracy, Precision, Recall and F1-Score respectively. Across all attacks, accuracy maintains high values, with the best model maintaining between average accuracy between 99% and 100%. The average accuracy values were found to be 99.25%, 99.15%, 99.59%, 99.75%, and 99.61% for the SVM, KNN, DT, RF, and VE respectively. The standard deviation of accuracy values was found to be 1.08%, 1.06%, 0.58%, 0.32% and 0.45% for each of the models.

| Attack | SVM | KNN | DT | RF | VE |
|---|---|---|---|---|---|
| **1** | 99.80 | 99.84 | 99.94 | 99.96 | 99.88 |
| **2** | 98.94 | 99.26 | 99.61 | 99.74 | 99.43 |
| **3** | 99.89 | 99.65 | 99.93 | 99.93 | 99.89 |
| **4** | 95.30 | 95.54 | 99.64 | 99.78 | 99.32 |
| **5** | 99.86 | 99.86 | 99.95 | 99.97 | 99.92 |
| **6** | 99.71 | 99.57 | 99.68 | 99.86 | 99.80 |
| **7** | 99.93 | 99.81 | 99.93 | 99.97 | 99.93 |
| **8** | 99.26 | 98.44 | 99.50 | 99.83 | 99.45 |
| **9** | 99.88 | 99.96 | 99.99 | 100 | 100 |
| **10** | 99.32 | 98.92 | 99.34 | 99.46 | 99.30 |
| **11** | 99.85 | 99.84 | 99.92 | 99.96 | 99.94 |
| **12** | 99.85 | 99.85 | 99.90 | 99.89 | 99.87 |
| **13** | 99.59 | 99.59 | 99.31 | 99.62 | 99.62 |
| **14** | 99.99 | 99.92 | 99.93 | 99.99 | 99.99 |
| **15** | 98.80 | 98.12 | 99.06 | 99.32 | 98.95 |
| **16** | 99.02 | 99.21 | 99.75 | 99.88 | 99.73 |
| **17** | 97.76 | 97.90 | 97.39 | 98.27 | 98.17 |
| **18** | 99.98 | 99.92 | 99.97 | 99.99 | 99.98 |
| **19** | 99.00 | 98.63 | 99.44 | 99.45 | 99.51 |
| **Average** | **99.25** | **99.15** | **99.59** | **99.75** | **99.61** |

Table 3.6: Accuracy By Attack

| Attack | SVM | KNN | DT | RF | VE |
|---|---|---|---|---|---|
| **1** | 100 | 99.69 | 99.41 | 100 | 100 |
| **2** | 99.52 | 98.58 | 97.10 | 99.95 | 99.95 |
| **3** | 99.28 | 99.62 | 99.33 | 99.69 | 99.69 |
| **4** | 94.66 | 90.24 | 97.23 | 99.49 | 99.49 |
| **5** | 99.17 | 100 | 99.49 | 100 | 100 |
| **6** | 98.28 | 99.72 | 96.48 | 99.79 | 99.84 |
| **7** | 99.34 | 99.84 | 99.69 | 99.95 | 99.85 |
| **8** | 97.10 | 99.02 | 96.61 | 99.69 | 99.43 |
| **9** | 99.29 | 99.67 | 99.78 | 100 | 99.95 |
| **10** | 96.57 | 97.27 | 94.80 | 97.69 | 97.18 |
| **11** | 96.00 | 94.29 | 92.11 | 98.23 | 95.29 |
| **12** | 99.90 | 99.74 | 98.93 | 99.90 | 100 |
| **13** | 99.98 | 99.81 | 97.69 | 99.90 | 99.98 |
| **14** | 100 | 100 | 99.83 | 100 | 100 |
| **15** | 98.54 | 98.23 | 97.30 | 98.88 | 98.88 |
| **16** | 95.27 | 96.62 | 98.98 | 99.50 | 98.86 |
| **17** | 99.49 | 95.95 | 74.57 | 88.97 | 99.19 |
| **18** | 99.95 | 99.87 | 99.87 | 99.97 | 99.95 |
| **19** | 96.58 | 98.25 | 97.33 | 98.99 | 98.77 |
| **Average** | **98.36** | **98.23** | **96.65** | **98.98** | **99.28** |

Table 3.7: Precision By Attack

The precision results maintain high levels of performance, with average precision values of 98.36%, 98.23%, 96.65%, 98.89%, and 99.28%. Precision is the first metric which a distinct difference between models can be seen. The Decision Tree model performs the worst by

about 2% on average. While 17 remained as a consistent poor performer from accuracy, 11 was seen to have poor performance as well in this attack. While Random Forest had the best average precision, it performed poorly within attack 17, scoring more than 10% worse than SVM.

| Attack | SVM | KNN | DT | RF | VE |
|---|---|---|---|---|---|
| **1** | 96.15 | 97.38 | 99.46 | 99.27 | 97.81 |
| **2** | 80.42 | 87.28 | 95.50 | 95.16 | 89.26 |
| **3** | 98.47 | 93.45 | 99.23 | 98.87 | 98.06 |
| **4** | 87.32 | 13.11 | 95.73 | 96.03 | 94.88 |
| **5** | 98.06 | 97.14 | 99.59 | 99.34 | 98.47 |
| **6** | 95.84 | 91.68 | 97.16 | 97.41 | 96.20 |
| **7** | 99.19 | 96.34 | 98.88 | 99.39 | 98.83 |
| **8** | 88.29 | 70.39 | 93.49 | 96.92 | 92.87 |
| **9** | 98.26 | 99.46 | 100 | 100 | 100 |
| **10** | 89.61 | 80.62 | 91.83 | 91.26 | 88.59 |
| **11** | 57.14 | 52.38 | 83.33 | 88.10 | 81.72 |
| **12** | 97.10 | 97.36 | 99.03 | 97.97 | 97.41 |
| **13** | 97.06 | 97.27 | 97.70 | 97.39 | 97.29 |
| **14** | 99.90 | 99.42 | 99.67 | 99.92 | 99.90 |
| **15** | 92.79 | 88.18 | 95.93 | 96.26 | 93.57 |
| **16** | 97.69 | 97.61 | 99.17 | 99.60 | 99.22 |
| **17** | 57.40 | 62.48 | 76.06 | 76.45 | 65.51 |
| **18** | 99.80 | 99.33 | 99.85 | 99.92 | 99.85 |
| **19** | 92.84 | 87.21 | 96.74 | 95.24 | 96.03 |
| **Average** | **90.70** | **84.63** | **95.77** | **96.03** | **93.97** |

Table 3.8: Recall By Attack

| Attack | SVM | KNN | DT | RF | VE |
|---|---|---|---|---|---|
| **1** | 98.04 | 98.52 | 99.43 | 99.63 | 98.89 |
| **2** | 88.95 | 92.59 | 96.29 | 97.50 | 94.30 |
| **3** | 98.87 | 96.44 | 99.28 | 99.28 | 98.87 |
| **4** | 72.11 | 22.90 | 96.48 | 97.73 | 97.13 |
| **5** | 98.61 | 98.55 | 99.54 | 99.67 | 99.22 |
| **6** | 97.05 | 95.54 | 96.82 | 96.82 | 97.99 |
| **7** | 99.26 | 98.06 | 99.28 | 99.67 | 99.33 |
| **8** | 92.49 | 82.28 | 93.68 | 96.51 | 96.04 |
| **9** | 98.77 | 99.57 | 99.89 | 100 | 99.97 |
| **10** | 92.96 | 88.16 | 93.29 | 94.36 | 92.68 |
| **11** | 71.64 | 67.34 | 71.48 | 72.45 | 87.98 |
| **12** | 98.48 | 98.53 | 98.98 | 98.92 | 98.69 |
| **13** | 98.50 | 98.52 | 97.50 | 98.63 | 98.62 |
| **14** | 99.95 | 99.71 | 99.75 | 99.96 | 99.95 |
| **15** | 95.58 | 92.93 | 96.61 | 97.55 | 96.14 |
| **16** | 96.46 | 97.11 | 99.08 | 99.55 | 99.04 |
| **17** | 72.80 | 75.68 | 75.31 | 82.24 | 78.91 |
| **18** | 99.87 | 99.61 | 99.86 | 99.95 | 99.90 |
| **19** | 94.67 | 92.40 | 97.04 | 97.08 | 97.38 |
| **Average** | **92.89** | **89.18** | **95.26** | **96.18** | **96.37** |

Table 3.9: F1-Score by Attack

The average recall was found to be 90.70%, 84.63%, 95.77%, 96.03%, and 93.97% for each model. Attacks 11 and 17 maintain the worst performance within attacks. Notably, KNN has an outlier of 13.11% for attack 4, which reduces its average by 3% and increases its standard deviation by 7%, when compared to the rest of the results for KNN without it. Within the weak attacks, for attack 11, RF is able to maintain a high value of 88.17% despite the rest of the models performing poorly.

The average F1-Scores were found to be 92.89%, 89.18%, 95.26%, 96.18% and 96.37% respectively. Since the F1-Score is the harmonic mean of precision and recall, the values fall between the two calculated values, and is often used as an overall performance evaluator of models.

### 3.5.1 Misbehavior Value Modification

After training, an evaluation of the results was completed to determine any significant outliers. While there were other attacks which performed poorly, the DoS and Message Replay attacks performed worse than would be expected for those attacks. While evaluating the DoS attack data, an inconsistency in logic used within F2MD was detected. At some times a DoS attack would be indicated when time between messages was the same as normal, and other times the time between messages would be smaller while not being marked for a DoS attack. It is believed that this occurs due to misbehaving vehicles sending out messages not marked as misbehaviors at the message sending rate of normal vehicles, regardless of if the vehicle was in the middle of a DoS attack. However, even if the normal messages are being sent out at a normal interval, when compared to the time between messages sent from the attacking vehicle, they would be regarded as abnormal time differences. In order to rectify this logic error, all messages were marked as misbehaving if they sent messages at time difference of less than 60% the normal one. This value was chosen as the DoS attack was implemented to send messages at 50% and 25% the normal time delay, so 60% encompasses both while allowing a significant error range for both misbehaving and normal messages. A comparison of average results across all models before and after this change can be seen in Table 3.10.

| | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
|---|---|---|---|---|
| **Unmodified** | 94.83 | 92.43 | 69.05 | 78.43 |
| **Modification** | 98.96 | 98.76 | 94.59 | 96.48 |

Table 3.10: Change in DoS Results

The second attack to be evaluated was the data replay attack. Within this attack, a vehicle would first function as a normal vehicle, while collecting messages being sent from another vehicle. It would then transition into attack mode, and instead of sending BSMs including its own information, it would instead forward the data from the collected messages. The

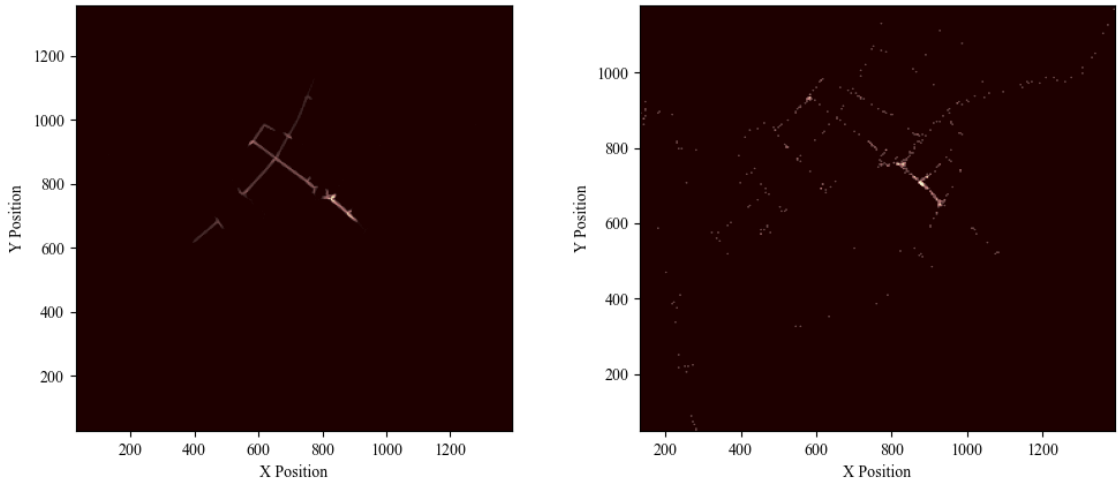
difference between this attack and others completed is that after the swap, there is not any difference of behavior between this vehicle and legitimate ones. Within the data however, each follow up message is marked as an attack. In order to best represent how this attack would be identified, a change was made such that only the first message marked as an attack from a vehicle was kept as being marked as an attack. The results before and after this change can be seen below in Table 3.11.

| | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
|---|---|---|---|---|
| **Unmodified** | 97.28 | 81.38 | 59.33 | 68.90 |
| **Modification** | 99.89 | 95.73 | 73.276 | 83.01 |

Table 3.11: Change in Replay Results

The replay attack is a good case study of the differences between a centralized IDS and a localized one. As mentioned previously, sequence length is the primary benefit obtained from a centralized IDS, while a localized node is able to better train to its location. While detecting replay attacks, a long sequence is able to be evaluated to locate the change in behavior of a vehicle over a longer period of time. This means that if the IDS marks a single sequence as non-malicious, it will still have $sequencelength - 1$ more chances to appropriately detect the change in behavior. In addition, there is able to be more context for behavior before and after the change, and it should be easier to detect the jump because of that. On the other hand, an RSU-Based IDS is able to have a more accurate understanding of what should be occurring within its area due to reduced input complexity. Figure 3.6 shows heatmaps of normal and misbehaving positions to demonstrate this.

Because the RSU is able to specialized in a single location, even without the context which comes from the sequence length, the RSU is able to immediately able to detect a replay attack as it enters the RSU range because of its difference from the typical first messages the RSU will receive. In addition, the RSU is still able to notice a change in behavior if the




(a) Behaving Vehicle Positions (b) Misbehaving Vehicle Positions

Figure 3.6: Positional Data For a Replay Attack

attack starts within its range. While our implementation only utilizes a sequence length of two, meaning there would be only one chance to detect the behavior, longer sequence lengths could be employed within RSU-Based IDS systems.

### 3.5.2 Precision-Recall Trade-Off

In situations where accuracy is never able to reach 100%, there will a trade-off between the precision and recall identification rates. Imperfect accuracy means that there will be miss-classifications occurring, which precision evaluates through considering the False Positive Rate and Recall evaluates through considering False Negative Rate. As such, it is important for a system to determine if it is more important to reduce the number of innocent vehicles being marked as malicious, precision, or if it is more important to ensure that no attack goes undetected, recall. An example of this trade-off is shown in Figure 3.7.

While this trade-off can be complex to determine what is best for a system, for the current VANET network architecture, it seems likely that recall is the more important factor to

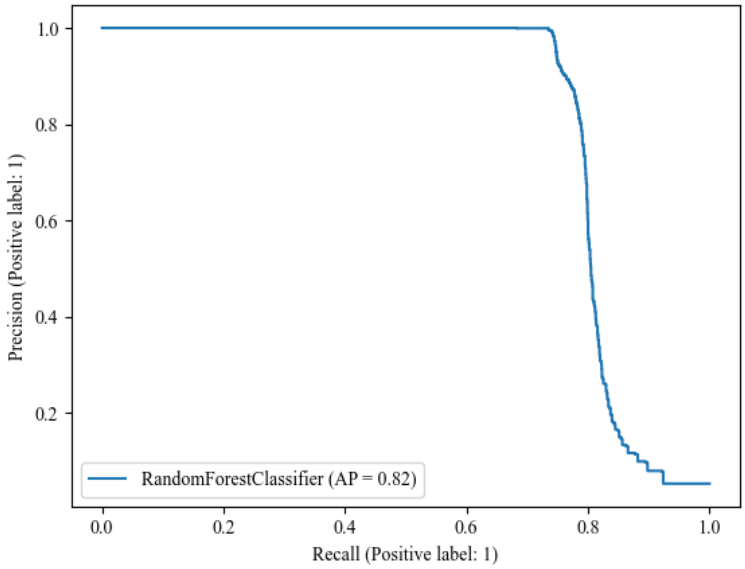



Figure 3.7: RF Precision-Recall Curve for Traffic Congestion Sybil

consider. The usage of a GMBD at the highest point of the architecture is used to evaluate report to determine the accuracy of reports, as such it is more important to ensure that the the reports make it to the GMBD in the first place. In addition, the extra annoyance faced by drivers when labeled with a False Positive is less detrimental to not identifying a cyberattack and taking an action off of that message.

### 3.5.3 Model Execution Time

The final metric evaluated is the execution time of each model to compute the results for any single data point. The testing of the models was conducted on a 5950x with 16GB of RAM. While this computing power is higher than what would be expected for current RSU implementations, it is unclear what the level of computing power will be when RSUs are actually deployed. In addition, the comparison of values between models should remain fairly consistent regardless of the computing power given. Each of the models executed in a

different order of magnitude, with KNN taking 1.6E-3 seconds, SVM taking 2.1E-4 seconds, RF taking 5.6E-6 seconds and DT taking 1.4E-7 seconds. The VE takes slightly longer than the sum of the rest of the models, at 1.9E-3 seconds, as not only does it need to evaluate each model, but it also incurs the overhead of the voting.

| Model | ET (ms) |
|---|---|
| **Support Vector Machine** | 0.21 |
| **K-Nearest Neighbors** | 1.68 |
| **Decision Tree** | 0.00014 |
| **Random Forest** | 0.0056 |
| **Voting Ensemble** | 1.91 |

Table 3.12: Average Execution Time by Model Type

The number of cars which can be in an RSUs range before it become overloaded can then be calculated. At a message rate of 0.1 HZ as suggested by SCSM, and assuming that the RSU is able to transition between evaluations without any overhead and that the voting overhead remains constant for different numbers of models in the VE, the number of vehicles an RSU can maintain within its 300m radius before being overloaded was then calculated and displayed in Table 3.13.

| Model | Vehicle Capacity |
|---|---|
| **VE(KNN, SVM, RF, DT)** | 52 |
| **VE(SVM, RF, DT** | 434 |
| **VE(RF, DT)** | 5,000 |
| **Random Forest** | 17,857 |

Table 3.13: Vehicle Capacity For Each Model Type

## 3.6 Discussion

A comparison with previous works was conducted in Table 3.14. This comparison includes any work which has considered 4 or more attacks, and takes the values for each metric from

the model that performed best for that metric in that work. Our models can be seen to achieve the highest level of accuracy yet, and extremely high precision values and recall values as well. Within the literature, a trend of traditional machine learning algorithms having higher precision accuracy and precision values, and lower recalls has been shown through both [62] and our own work. Deep Learning algorithms have been shown to mitigate this reduction in recall, at least slightly, as shown in [3]. Deep learning usage within our RSU IDS model can be explored more in the future.

| | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
|---|---|---|---|---|
| [**67**] | 95.90 | - | - | - |
| [**5**] | 99.61 | - | - | - |
| [**62**] | 90.83 | 99.17 | 66.28 | 69.84 |
| [**2**] | 98.0 | 99.6 | 95.6 | 97.6 |
| [**3**] | 99.65 | 98.26 | 98.26 | 98.26 |
| **Our Model** | 99.75 | 99.28 | 96.03 | 96.37 |

Table 3.14: Comparison with Previous Works

This paper investigated the usage of location specialization, which has been shown to produce positive results through our high accuracy and precision. Because RSUs are situated in a single location for a long period of time, neither Vehicle-Based IDSs or Cloud-Based IDSs will be able to take advantage of location specialization due to the large locational areas that they will need to cover. [3] proposes the usage of edge-computing instead of cloud computing. If deployed correctly, and the ratio of RSUs to edge devices is kept low, edge devices could acquire the ability to specialize, albeit in a larger area reducing the benefit from location specialization. However, as SCSM requires the disinfection of reports being sent to the GMBD to prevent location tracking, it seems unlikely that SCMS would allow the transmission of BSMs themselves without the removal of identifying information. As such, cloud devices and edge computing devices could end up unable to utilize location specialization, unless an exemption or adjustment is able to be made. Any exemption or

adjustment would require a reduction in privacy.

Finally, a comparison of the system architectures can be completed between those located in Figures 3.1 and 3.2. The LMBD to GMBD can first be dismissed. Due to being located in constrained systems, LMBD is expected to be unable to run complex algorithms, as well as being unable to perform location specialization. The second option within Figure 3.1 can be evaluated for an IDS location of either edge computing or cloud servers. Both of these options require further development and deployment of system components, which will cost money and potentially introduce undue complexity into an already complex VANET security system. Cloud computing is unable to utilize location specialization, while edge computing may be able to perform location specialization at the cost of privacy. And while wired data transmission may not be a major concern, based on 2015 data, American drivers spent over 84 billion hours driving, which at a BSM frequency of 10Hz and a BSM size of 200 bytes, is equivalent to sending over 600 petabytes of vehicle data per year [15]. This is a staggering amount of data which would need to be sent to the IDSs, and this only includes driving data from the USA.

Our RSU-Based model is able to utilize location specialization, does not incur any further cost as it is a part of the other network topologies, and will only need to send generated reports instead of every BSM. The usage of location specialization is able to increase IDS performance by being able to compare incoming messages to typical messages recieved within its specialized area. It reduces system complexity, by fitting into the existing SCMS structure, and reduces implementation and installation costs by not requiring other components like edge computing devices or servers. Since it only needs to send out reports, it could prevent the sending of a potential 600 petabytes of vehicle data.

## 3.7 Conclusion

This paper has introduced an RSU-Based IDS system which uses a hybrid LMBD/consolidation node. This system is able to use location specialization unique to an RSU to improve IDS performance by reducing input complexity. The system also eliminates the need for the planning and installation of additional infrastructure into the VANET system, such as cloud or edge computing. This can reduce implementation and installation costs as well as reduce system complexity. Due to the elimination of external infrastructure, it is able to remove the need to forward every BSM recieved, which could reduce the amount of forwarded information by up to 600 petabytes of vehicle data as well as increase privacy.

This paper is the first to implement an RSU-Based IDS within a standardized data set or framework. It modifies the F2MD framework to generate a data set comparable to that of the VeReMi Extension data set, only using RSUs instead of vehicles. An evaluation of this data set was then completed using Support Vector Machines, K-Nearest Neighbors, Decision Trees, Random Forest, and a Voting Ensemble of the previous four models. These models were able to perform at a high level, creating the highest ever average accuracy across all attacks of 99.75%. The models were also able to create high precision values of 98.23-99.28% and good recalls of 95% and 96%.

# Chapter 4

# Intrusion Response within an Autonomous Intersection Controller

## 4.1 Abstract

*Abstract-* Vehicle Ad-hoc Networks (VANET) and Connected and Autonomous Vehicles (CAV) have the ability to drastically improve the safety and efficiency of our roadways. As a cyber-physical system, it is of critical importance to prevent, detect and react to cyber-attacks and threats, as a compromised system could cause real world safety concerns. While significant research has been done into prevention and detection of cyberattacks, the reaction aspect is often overlooked. This paper considers the reactions of an adaptive autonomous intersection under a sybil attacker. A wireless signalized intersection mode is introduced into the autonomous intersection and an evaluation of the intersection is conducted. The adaptive intersection boasts a 78% decrease in delay while under attack over one without defenses.

## 4.2 Introduction

As the usage of CAVs and VANETs become closer to a reality, the potential for greater efficiency and safety within transportation systems has drawn in an increasing large amount of research. However, before any application which increase safety or efficiency can be used, the security of VANETs and the applications themselves must be ensured.

As Cyber-Physical Systems (CPS) VANETs have increased security and design considerations which typical cyber-only networks do not need to consider. Despite the increased complexity, the required behavior of VANET security can still be categorized using the traditional three step process of prevention, detection and response [18, 25]. The implementation of these steps is often referred to as Intrusion Prevention Systems (IPS), Intrusion Detection Systems (IDS), and Intrusion Response Systems (IRS). [69] details the importance of addressing concerns at design time rather than after deployment.

[23] breaks the security concerns of the CPS down into the cyber and physical systems separately. They identify confidentiality, integrity, availability, authentication and non-repudiation as important aspects for the cyber system and deterrence, detection, delay, response and neutralization as important aspects for the physical system. This paper details the relationships between cyber requirements and physical requirements by detailing the relationships as non-existent, mutually beneficial, or supporting. [18] further details IRS options, specifically when dealing with sensing units such as those common in IoT systems. They suggest the usage of statistical analysis to ignore data sent by compromised systems and to plan the system with enough redundancy to be able to function even if half of the system becomes compromised. They also discuss the usage of game theory and event-triggering controls to counter DoS attacks.

As mentioned, the one of the primary goals of VANETs is to facilitate applications which

improve efficiency and safety. Safety applications typically use communication to forward safety information to vehicles which may not have access to it due to their location within a traffic flow. Efficiency applications on the other hand are designed to improve one or more aspects of the Intelligent Transportation System (ITS) that the CAVs are located in. Energy usage [76] and communication routing [36, 55, 74] have both been looked at deeply.

The final efficiency application type we will look at is one that controls vehicle pathing and traffic flow. Within this application type, vehicle pathing provides more general paths for a vehicle to follow, such a directing it to take specific links. [28, 31, 40] have all examined different ways to optimize vehicle paths. However, some applications look to control vehicles at a more microscopic level. These application types have direct control over the movement of a vehicle, and as such, despite not being safety applications, are safety critical. Two common applications like this are Cooperative Adaptive Cruse Control (CACC) and autonomous intersections.

This paper introduces the concept of a willfully compromised system for usage within safety critical applications. We model the interaction between a sybil attacker and the IRS of an autonomous intersection, and then evaluate the impact of a resilience mode within the autonomous intersection.

## 4.3 Literature Review

IPSs, IDSs, and IRSs play a critical role within any network for continued security. Within VANETs, IPSs and IRSs have been explored, but the primary focus in this field has been the study of IDSs. Attacker models, trust scores and priority detection have all been shown to be efficient IPSs [9, 58, 72].

There have been a variety of ways that IDSs have been modeled and trained. [57] utilized a hierarchical IDS structure similar to that used within the Security Credential Management System (SCMS). This paper uses a with local misbehavior detection within vehicles to generate reports and submit them to a global authority. In this case, the global authority is a vehicle elected as the cluster head using a Trust-Based system. [30] used a similar system which used a central server as the global authority.

Trust and reputation based systems have been utilized within both IDSs and IPSs [39, 42, 50]. However, the usage of pseudonyms within VANETs has created issues with tracking the trust throughout a system. Some applications have tracked vehicles through their kinematics instead of through vehicle ids, but there are privacy concerns about circumventing the privacy afforded by pseudonyms. Reinforcement learning, machine learning and deep learning have also all been utilized [2, 56, 62] within IDSs. [38] proposed a Bayesian game theory approach which utilizes a deep learning model.

Intrusion Response Systems are the least studied of the security systems. The most common implementation of IRSs are those which do not make any decisions. IRSs often will receive a positive identification for misbehavior or intrusion into a system from IDSs, and then will revoke the security certificates. [20] describes the process which SCMS uses to complete this action in order to revoke all pseudonyms for the vehicle. This approach is one which is often used within other networks outside of VANETs. However, as a result of being a CPS, VANETs do not have the option to simply ignore an attacker and believe that it will not impact the system.

[41] provides an IRS within a vehicle for a DoS attack. They state that even if a DoS attack is recognized to be occurring, there will still be an impact on the network and vehicles should take appropriate actions to avoid that part of the network. They propose a game which allows vehicles the decision to continue, change directions, or stop for a period of time to

get out of the range of the DoS attacker. [59] uses a hierarchical game which introduces an Intrusion Decision Agent (IDA) which controls a IPS, IDS, and IRS. Within this system, the IRS has the ability to revoke certificates and take actions to change a vehicle's pseudonym or cryptographic key to avoid tracking and eavesdropping.

Within IRSs there are two critical metrics used by IDSs which need to be considered: the Detection Rate (DR) and the False Positive Rate (FPR). DR determines the percentage change that an attack will be properly identified as an attacker, while FPR determines the chance that an identification of an attack was actually caused by an attacker. While DR is important to consider to ensure that a system is able to actually respond to an attack, FPR is critical to understand to ensure that in the event of a false positive, a safety-threatening action is not taken. Within a CACC system, if messages designating a vehicle's position in the platoon are determined by the system to be incorrect, cars speeding up to take its place could cause a crash if the identification is a false positive.

Autonomous intersection control has traditionally been defined using three states: Centralized Intersection Management (CIM), Distributed Intersection Management (DIM) and Wireless Intersection Failure (WIF) [17]. The CIM utilizes a centralized RSU as the manager for the intersection. A vehicle will send a request to the manager for access to the intersection, to which the RSU will make a new reservation and potentially adjust already existing reservations. The RSU will then respond to the vehicle as well as any other vehicles whose reservations have updated with the access information required for the intersection.

DIM operates similarly to CIM, however it needs to accomplish the same tasks without a centralized hub. DIM can either elect a temporary manager, which will pass off the responsibility after it moves through the intersection, or will perform management communally. The usage of blockchain in [12] is one example of communal management. Although DIM is able to optimize routing through an intersection as well as CIM, DIM is typically seen

as being unable to incorporate system-wide approaches which are able to improve efficiency outside of a single intersection.

WIF is demonstrated through an intersection falling back to a four way stop. This is often used to demonstrate the inefficiency caused by not using communication to manage an intersection. Table 4.1 contains information on the benefits and requirements needed for each state, with Vehicle-To-Everything (V2X) communication and Infrastructure-To-Everything (I2X) being used to show the network state.

| | **CIM** | **DIM** | **WIF** |
|---|---|---|---|
| Description | Routing is coordinated through a centralized hub in an RSU | Routing is coordinated through various means, either communally or through a managing car | Intersection transitions to a four-way stop |
| Requirements | V2X, I2X, and GPS data | V2X and GPS data | Sensor Data |
| Benefits | Most Efficient Form, Able to optimized for different parameters | Second most efficient form, Able to optimize for different parameters, RSU not required | Low reliance on communication or single sensors working correctly |
| Detriments | Heavy reliance on many systems working | Heavy reliance on a few systems working | Significantly less efficient then other options, reduction in safety |

Table 4.1: Autonomous Intersection State Information

CIM and DIM both have the option to perform optimization within the reservation system of an intersection. The way that vehicles and intersections are modeled has a significant impact on how the optimization is able to occur and the method employed for the optimization. [21] uses a spacial discretization optimization to book reservations. [73] utilizes a time sequence based optimization and Petri Net optimization to reduce delay. [75] utilizes path trajectories to perform optimization by minimizing spaces between paths. [35] optimizes intersection reservations through the use of a Phase Conflict Map. [11] uses Ant-Colony optimization within a CIM.

## 4.4 Methodology

This paper implements an Intrusion Response System which is utilized within an autonomous intersection. As stated before, although intersection management is not specifically a safety

application, the results of the application are safety critical. Because of the risk posed by false positive identifications within the IDSs of safety critical applications, we introduce the concept of a Willfully Compromised System (WCS). A WCS is one which recognizes that there is a high chance that a vehicle or message is an attack or misbehavior, but will still allow the vehicle or message to impact the system in order to not cause any safety issues if the detection is a false positive. Within a CACC application, this might involve leaving a gap in the platoon which an attack states a vehicle is located in. Within an autonomous intersection, this would involve allowing a vehicle which does not exist to make a reservation for the intersection, and potentially leave the intersection underutilized for a period of time.

These applications will enter into a WCS state upon the detection of an attack or misbehavior, and will remain in that state until a stopping criteria is reached. This criteria may be the dissolution of the platoon, the attacker passing through the intersection, or a certificate revocation followed by a re-initialization of the application without the WCS state.

### 4.4.1 Autonomous Intersection

The autonomous intersection operates as a CIM by using an RSU present near the intersection to direct incoming vehicles. It is assumed that outside of the RSU there is no other infrastructure available at the intersection. The RSU will send out frequent beacons, and any vehicles which have just entered the range of the RSU will respond with a Reservation Request. The RSU will collect all responses sent between the beacons, make the reservations, and send the reservation responses in the next beacon message.

Upon the detection of an attack or misbehavior, the RSU will enter a WCS state. As discussed earlier, CIM, DIM, and WIF each have specific requirements for communication and data needed for the system's appropriate use. However, within the WCS state, the V2X

information that is being received may not be seen as trustworthy. As such, a new Wireless Signalized Intersection (WSI) state is introduced to improve resiliency when under attack.

Within WSI, even without the ability for V2X communication or GPS data, the intersection can fall back to becoming a signalized intersection which is run by the RSU. While there are not many significant optimizations which can be made to WSI, preventing the breakdown to Wireless Intersection Failure means that heavily trafficked areas are able to continue their normal operations and prevent long queues from forming, even if there is a decrease in efficiency. The following system requirement chart was created to show the demands for each of the intersection states.

| | CIM | DIM | WSI | WIF |
|---|---|---|---|---|
| I2X Communication | ✓ | X | ✓ | X |
| V2X Communication | ✓ | ✓ | X | X |
| GPS Data | ✓ | ✓ | X | X |

Table 4.2: Autonomous Intersection State Requirements

### 4.4.2 Attacker-Intersection Interaction

The attacker in this paper is a corrupted vehicle which creates a sybil attack on the intersection by sending multiple requests to the intersection using its pseudonyms. As detailed in the SCMS protocol, an attacker has a minimum of 20 pseudonyms which it can use as attacks. The attacker is assumed to be rational, and is not attempting to cause physical harm to itself or to other vehicles. Instead, it is only interested in decreasing the efficiency of the autonomous intersection through its sybil attacks.

After entering the RSUs range, the attacker will first make a reservation for itself within the intersection. It will then determine the number of attacks, $a_n$ which it will conduct. The attacker will attempt to cause as large of a delay as possible, while using as few pseudonyms,

or attack messages, as possible.

The intersection will act normally until the IDS identifies an attack, after which it enters into a WCS state. After entering into this state, upon evaluating each message it will have the option to either accept the reservation, or to transform from CIM to WSI. The intersection will make decision in order to minimize the delay occurring from the intersection.

### 4.4.3 Simulation

This simulation was built in the VEINS framework, which is a combination of OMNeT++ and SUMO [65]. This simulation is structured around a 4-way single lane approach intersection.

The CIM created is hosted within the RSU and collects responses from vehicles entering into the range of its RSU beacon. A 1:8:1 ratio of left turn, through traffic and right turn was used. The CIM uses sixteen critical points within the intersection to make reservation for vehicles passing through.

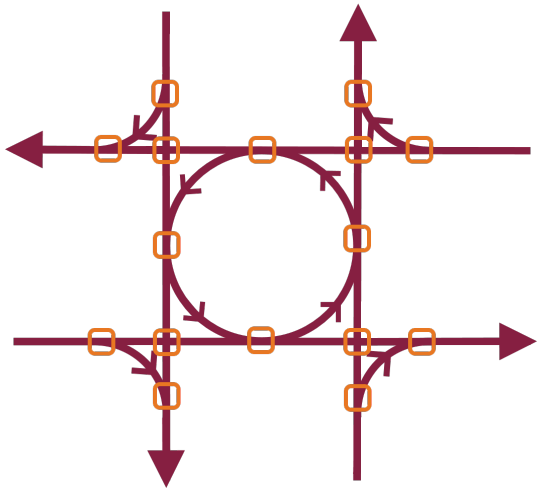


Figure 4.1: Autonomous Intersection Critical Points

The RSU will receive reservation requests from vehicles, including information such as their Pseudonym ID, Location, Speed, Desired Path and Time Sent of Request Message. It will then reserve time slots for each of the critical points the vehicle will pass through, giving a buffer time of 0.5 seconds between vehicles for each critical point. The RSU then responds with the time that a vehicle will need to delay before it reaches the first critical point. The RSU provides access to the intersection on a First-Come-First-Serve (FCFS) basis.


Figure 4.2: Simulated Intersection Overview

After entering the WCS state, the intersection will continue to make reservations until the delay that is incurred by allowing the potentially non-existent vehicles to pass through the intersection is greater than the delay which would be incurred in utilizing the WSI mode. Upon swapping to the WSI mode, the intersection will determine which reservations need to be completed in order to maintain safety, and then instruct all other vehicles to come to a stop at the intersection. After the last of these crossing vehicles passes through the

intersection, the RSU will begin to use WSI functionality. It will use a 40 second green time, 4 second yellow time, 40 second green time, and 4 second yellow time. The lanes with the highest level of traffic will be sent first. Once all vehicles that were in range when the WCS state was triggered have moved through the intersection, the intersection will leave the WCS state and transition from WSI back to CIM.

## 4.5 Results

The autonomous intersection, as well as the Intrusion Response System, was constructed and evaluated. The primary evaluation metric used was vehicle delay through the intersection. To provide an evaluation of the performance of the autonomous intersection, a comparison was conducted between it, a four way stop and an intersection using the same parameters as the Wireless Signalized Intersection mentioned above. The average delay for each was evaluated against increasing vehicle flow rates. The flow rates were measured as vehicle approaching the intersection per second. Flow rates from 0 to 0.7 vehicles per second (v/s) with a step size of 0.025 were evaluated.

As can be seen in Table 4.3 the autonomous intersection significantly outperformed both the four-way stop and the signalized intersection. The four way stop data points were removed from the graph after 0.45 v/s as their values increased to be over 50 seconds of delay. While the four way stop was initially more effective than the signalized intersection, the signalized intersection quickly overtook it as the flow rate increased. One primary cause of the over-performance of the autonomous intersection is its ability to quickly reset the state of the intersection. While variance can have a negative impact on a static signalized intersection discharge, the autonomous intersection is able to allocate usage of the intersection to the direction of flow which needs it the most. These results for the signalized intersection were

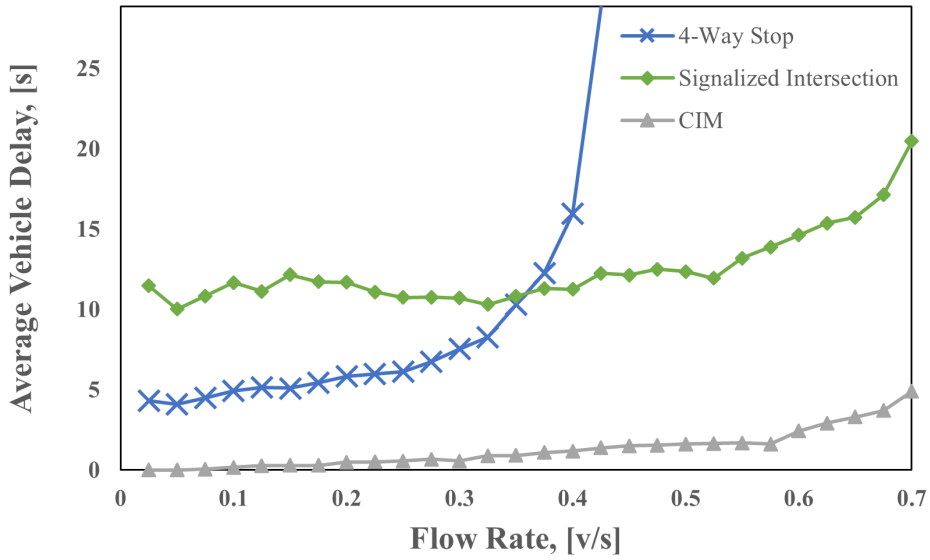



Figure 4.3: Average Vehicle Delay by Intersection Type

used for the delay needed to change from CIM to WSI.

The usage of the IRS is then explored while under a sybil attack. An intersection without an IRS is considered alongside an intersection which does contain one. The intersection without an IRS allows the attacker to make reservations and impact the traffic flow. The intersection with an IRS utilizes the Wireless Signalized Intersection as a fallback resiliency mode. A flow rate of 0.5 v/s was used to evaluate the intersection. A rolling average time of 84 seconds was chosen as the metric for comparison to align with a full cycle of the signalized intersection to ensure that there was not an unfair or beneficial light cast upon it.

Table 4.4 demonstrates the power of IRS and its usage of the WSI mode. The models maintained a normal status until the attacks were sent. At this point a clear difference can be seen. All of the delays immediately spike, however the intensity of the spike is seen to be largely mitigated by the IRS. The WSI can be seen to have a 78% decrease in average delay over the intersection with no defense during the duration of the attack, which in this case is between 200 and 350 seconds. The intersection with an IRS is able to recover faster,

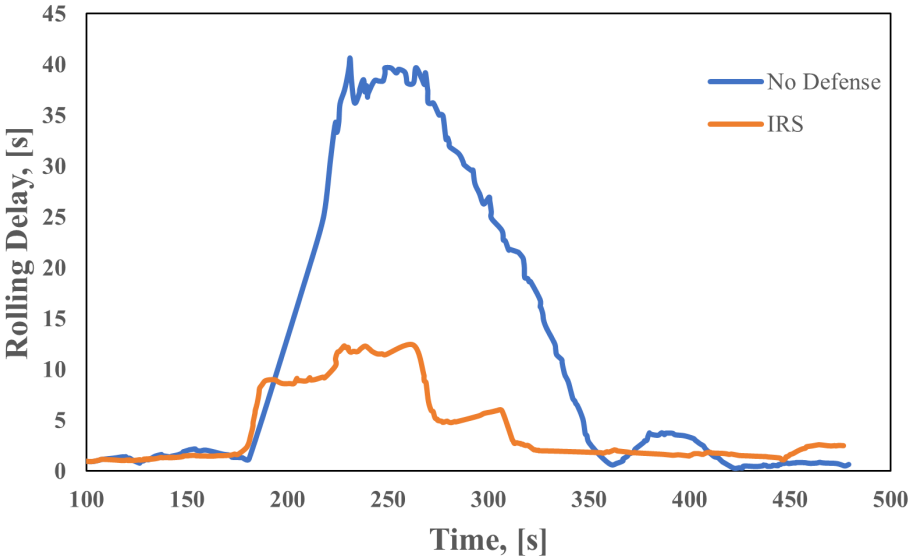



Figure 4.4: 84 Second Rolling Average Vehicle Delay Under Attack

due to reservations with large delay values not being allowed to remain in the autonomous intersection.

The final evaluation completed was the interaction which occurs between the attacker and the IRS. A sequence of attacks was executed against increasing flow rates through the autonomous intersection. For each flow rate, the minimum number of attacks needed to trigger a swap of intersection mode was evaluated.

A higher number of attacks were found to be needed to trigger a change at larger traffic volumes due to the more significant delays from the WSI. However, despite being harder to trigger, the higher traffic volumes remain impacted by the change for a significantly longer period of time. At higher volumes, more vehicles are approaching the intersection while the attack, or false reservations, are still being used. Because of this, the attack is able to chain together with legitimate approaching vehicles and cause delays for a longer period of time. Although the attacks were found to have increasing impacts at higher traffic volumes, the other systems also degraded due to the high traffic volumes, making a change more difficult

to trigger. The full results for this interaction can be viewed in Table 4.3.

| Flow Rate (v/s) | WSI Delay (s) | Number of Attacks |
|---|---|---|
| **0.1** | 11.66 | 5 |
| **0.2** | 11.68 | 5 |
| **0.3** | 10.71 | 4 |
| **0.4** | 11.25 | 5 |
| **0.5** | 12.36 | 6 |
| **0.6** | 14.63 | 7 |
| **0.7** | 20.46 | 9 |
| **0.8** | 22.09 | 9 |
| **0.9** | 32.89 | 10 |

Table 4.3: Number of Attacks Needed to Cause a Mode Switch

## 4.6 Conclusion

Although prevention and detection have been heavily researched within Vehicle Ad-hoc Networks (VANET), Intrusion Response Systems (IRS) have not had the same level of research conducted on them. The concept of a Wilfully Compromised System (WCS) has been successfully integrated into an IRS within an autonomous intersection.

In the future design of safety critical applications and infrastructure, the concept of a WCS should be fully evaluated and used. In the design of transportation systems, all attacks and countermeasures can and should be evaluated under the concept of a WCS. Within this paper, a defense against a sybil attacker was introduced. In addition to its ability to defend against cyberattacks using cyberdefenses, the WCS framework has the potential to be used fluidly within cyberphysical systems.

A Wireless Signalized Intersection (WSI) mode was introduced to the autonomous intersection as a resiliency measure to prevent a loss of efficiency. Swapping intersection modes to WSI was shown to have a 78% decrease in delay over a system without defenses.

# Chapter 5

# Conclusions

## 5.1 Conclusions

This thesis has explored the security of Vehicle Ad-hoc Networks (VANET). Both detection of and response to attacks and misbehaviors have been addressed throughout this paper. An Intrusion Detection System (IDS) within the F2MD framework was constructed. It implemented RSU-Based collection, training and testing of multiple Intrusion Detection Systems. Support Vector Machines, Random Forest, Decision Trees, K-Nearest Neighbors, and a Voting Ensemble were all evaluated using this data. These models were able to perform at a high level, creating the highest ever average accuracy across all attacks of 99.75%. The models were also able to create high precision values of 98.23-99.28% and good recall values of 95% and 96%.

The development of an Intrusion Response System (IRS) was constructed within an autonomous intersection facing a sybil attack. A Wireless Signalized Intersection (WSI) mode was introduced to the autonomous intersection as a resiliency measure. The usage of a Wilfully Compromised System (WCS) was also introduced within the paper. The system showed a 78% decrease in delay length during an attack over systems without these defenses.

## 5.2 Key Findings

The following Key Findings have been derived from the results of this thesis.

- The work completed in this thesis has shown positive results regarding the creation and utilization of realistic scenarios and data within simulations of Intelligent Transportation Systems (ITS). As more research has been completed within Intelligent Transportation Systems, more and more true-to-life systems will be able to be constructed without a substantial burden on the performance of the systems being tested.

- Chapter 3 utilized the F2MD framework to construct Roadside Units which were used to train and evaluate Intrusion Detection Systems. The usage of location specialization was able to improve the Intrusion Detection System's performance. The proposed system is able to eliminate the need for other excess infrastructure which reduces system complexity, implementation and installation costs, and the amount of transmitted data.

- Chapter 4 introduced a working Intrusion Response System within an autonomous intersection. The concept of a Wilfully Compromised System was introduced and used to maintain safe operations while showing increased performance over other alternatives. A Wireless Signalized Intersection was used as a resiliency measure within autonomous intersections to prevent a breakdown of safe intersection functionality.

## 5.3 Future Work

Through both of the sections of our work, there has been several areas of future work which have been discovered.

- There is a need to continue the study of an RSU-Based Intrusion Detection System. For our evaluation, sequence length was kept to a minimum in order to demonstrate the power of location specialization. Future research should be conducted to determine the optimal sequence length and to test deep learning models on the system.

- Further expansion of the F2MD framework should be completed. Although the VeReMi data set, which was built using the F2MD framework, is extremely helpful, the usage of only the data set limits the real world application of results. Construction of currently proposed networking frameworks within F2MD is suggested.

- Increased research into Intrusion Response Systems within Intelligent Transportation Systems is needed. Even if there is a large body of research being conducted within Intrusion Detection Systems, there is exceptionally minimal research done in Intrusion Response Systems. Even with high performances in detection, without an understanding of how systems will react to intrusion or misbehaviors, little can be accomplished. Further development of Wilfully Compromised Systems into other applications within Intelligent Transportation Systems could be one way to continue to grow this area of research.

- Further testing of a resilient autonomous intersection should be conducted. This thesis provides a solid foundation, but increased intersection size, integration into a network system, and a relaxing of the positive assumptions made on the attacker are all directions for future work to move in.

# Bibliography


[1] Mohammed Saeed Al-kahtani. Survey on security attacks in Vehicular Ad hoc Networks (VANETs). In *2012 6th Int. Conf. Signal Process. Commun. Syst.*, pages 1–9. IEEE, dec 2012. ISBN 978-1-4673-2393-2. doi: 10.1109/ICSPCS.2012.6507953.

[2] Tejasvi Alladi, Bhavya Gera, Ayush Agrawal, Vinay Chamola, and Richard Yu. DeepADV: A Deep Neural Network Framework for Anomaly Detection in VANETs. *IEEE Trans. Veh. Technol.*, PP(c):1, 2021. ISSN 19399359. doi: 10.1109/TVT.2021.3113807.

[3] Tejasvi Alladi, Varun Kohli, Vinay Chamola, F. Richard Yu, and Mohsen Guizani. Artificial Intelligence (AI)-Empowered Intrusion Detection Architecture for the Internet of Vehicles. *IEEE Wirel. Commun.*, 28(3):144–149, 2021. ISSN 15580687. doi: 10.1109/MWC.001.2000428.

[4] Diego Altolini, Vishwas Lakkundi, Nicola Bui, Cristiano Tapparello, and Michele Rossi. Low power link layer security for IoT: Implementation and performance analysis. *2013 9th Int. Wirel. Commun. Mob. Comput. Conf. IWCMC 2013*, pages 919–925, 2013. doi: 10.1109/IWCMC.2013.6583680.

[5] Goodness Oluchi Anyanwu, Cosmas Ifeanyi Nwakanma, Jae-Min Lee, and Dong-Seong Kim. Real-Time Position Falsification Attack Detection System for Internet of Vehicles. *IEEE*, pages 1–4, 2021. ISSN 19460759. doi: 10.1109/etfa45728.2021.9613271.

[6] Auto Alliance and Gloabla Automakers. 5.9 GHz DSRC Connected Vehicles for Intelligent Transportion Systems. Technical report, GloabAutomakers, 2013.

[7] Kamal Azghiou, Manal El Mouhib, and Abdelhamid Benali. Perspective on the reliability behavior of intelligent transport systems during the transition phase from legacy vehicles to autonomous and connected ones: Four-road intersections as a case study. *IEMTRONICS 2020 - Int. IOT, Electron. Mechatronics Conf. Proc.*, pages 21–26, 2020. doi: 10.1109/IEMTRONICS51293.2020.9216422.

[8] Gérard Biau and Erwan Scornet. A random forest guided tour. *Test*, 25(2):197–227, 2016. ISSN 11330686. doi: 10.1007/s11749-016-0481-7.

[9] Tarek Bouali, Hichem Sedjelmaci, and Sidi Mohammed Senouci. A distributed prevention scheme from malicious nodes in VANETs' routing protocols. *IEEE Wirel. Commun. Netw. Conf. WCNC*, 2016-September(Wcnc), 2016. ISSN 15253511. doi: 10.1109/WCNC.2016.7564928.

[10] Benedikt Brecht, Dean Therriault, Andre Weimerskirch, William Whyte, Virendra Kumar, Thorsten Hehn, and Roy Goudy. A security credential management system for V2X communications. *IEEE Trans. Intell. Transp. Syst.*, 19(12):3850–3871, 2018. ISSN 15249050. doi: 10.1109/TITS.2018.2797529.

[11] Khac Hoai Nam Bui and Jason J. Jung. ACO-Based Dynamic Decision Making for Connected Vehicles in IoT System. *IEEE Trans. Ind. Informatics*, 15(10):5648–5655, 2019. ISSN 19410050. doi: 10.1109/TII.2019.2906886.

[12] Alina Buzachis, Basilio Filocamo, Maria Fazio, Javier Alonso Ruiz, Miguel Angel Sotelo, and Massimo Villari. Distributed Priority Based Management of Road Intersections Using Blockchain. *Proc. - IEEE Symp. Comput. Commun.*, 2019-June:1159–1164, 2019. ISSN 15301346. doi: 10.1109/ISCC47284.2019.8969653.

[13] Joe Diether Cabelin, Paul Vincent Alpano, and Jhoanna Rhodette Pedrasa. SVM-based Detection of False Data Injection in Intelligent Transportation System. *Int. Conf. Inf. Netw.*, 2021-Janua:279–284, 2021. ISSN 19767684. doi: 10.1109/ICOIN50884.2021.9333942.

[14] Alex Castrounis. Machine Learning: An In-Depth Guide - Overview, Goals, Learning Types, and Algorithms, 2021. URL https://www.innoarchitech.com/blog/machine-learning-an-in-depth-non-technical-guide.

[15] US DOT Volpe Center. How Much Time Do Americans Spend Behind the Wheel?, 2017. URL https://www.volpe.dot.gov/news/how-much-time-do-americans-spend-behind-wheel.

[16] Varun Chandola. Anomaly Detection : A Survey. *ACM Comput. Surv.*, 2009.

[17] Lei Chen and Cristofer Englund. Cooperative Intersection Management: A Survey. *IEEE Trans. Intell. Transp. Syst.*, 17(2):570–586, 2016. ISSN 15249050. doi: 10.1109/TITS.2015.2471812.

[18] Michelle S. Chong, Henrik Sandberg, and Andre M.H. Teixeira. A tutorial introduction to security and privacy for cyber-physical systems. *2019 18th Eur. Control Conf. ECC 2019*, pages 968–978, 2019. doi: 10.23919/ECC.2019.8795652.

[19] Lara Codecá, Raphaël Frank, Sébastien Faye, and Thomas Engel. Luxembourg SUMO Traffic (LuST) Scenario: Traffic Demand Evaluation. *IEEE Intelligent Transportation Systems Magazine*, 9(2):52–63, 2017.

[20] Vehicle Safety Communications and Consortium Proprietary. Vehicle-to-Vehicle Communications Misbehavior Detection. *Veh. Saf. Commun. 6*, 6, 2019.

[21] Kurt Dresner, Peter Stone, Kurt Dresner, and Peter Stone. Traffic Intersections of the Future . Traffic Intersections of the Future ∗. *Twenty-First Natl. Conf. Artif. Intell. NECTAR Track (AAAI 06)*, pages 1593–1596, 2006.

[22] Secil Ercan, Marwane Ayaida, and Nadhir Messai. New Features for Position Falsification Detection in VANETs using Machine Learning. *IEEE Int. Conf. Commun.*, pages 0–5, 2021. ISSN 15503607. doi: 10.1109/ICC42927.2021.9500411.

[23] Glenn A. Fink, Thomas W. Edgar, Theora R. Rice, Douglas G. MacDonald, and Cary E. Crawford. Overview of Security and Privacy in Cyber-Physical Systems. *Secur. Priv. Cyber-Physical Syst.*, pages 1–23, 2017. doi: 10.1002/9781119226079.ch1.

[24] Abdallah R. Gad, Ahmed A. Nashat, and Tamer M. Barkat. Intrusion Detection System Using Machine Learning for Vehicular Ad Hoc Networks Based on ToN-IoT Dataset. *IEEE Access*, 9:1–1, 2021. ISSN 21693536. doi: 10.1109/access.2021.3120626.

[25] Amjad Gawanmeh and Ahmad Alomari. Taxonomy analysis of security aspects in cyber physical systems applications. *2018 IEEE Int. Conf. Commun. Work. ICC Work. 2018 - Proc.*, pages 1–6, 2018. doi: 10.1109/ICCW.2018.8403559.

[26] Fabio Goncalves, Bruno Ribeiro, Oscar Gama, Joao Santos, Antonio Costa, Bruno Dias, Maria Joao Nicolau, Joaquim MacEdo, and Alexandre Santos. Synthesizing Datasets with Security Threats for Vehicular Ad-Hoc Networks. *2020 IEEE Glob. Commun. Conf. GLOBECOM 2020 - Proc.*, 2020-Janua, 2020. doi: 10.1109/GLOBECOM42002.2020.9348149.

[27] IEEE 1609 Working Group. *IEEE Guide for Wireless Access in Vehicular Environments (WAVE) - Architecture*. IEEE, 2014. ISBN 9780738187563.

[28] Chang Guo, Demin Li, Guanglin Zhang, and Menglin Zhai. Real-Time Path Planning in Urban Area via VANET-Assisted Traffic Information Sharing. *IEEE Trans. Veh. Technol.*, 67(7):5635–5649, 2018. ISSN 00189545. doi: 10.1109/TVT.2018.2806979.

[29] Sohan Gyawali and Yi Qian. Misbehavior Detection using Machine Learning in Vehicular Communication Networks. *IEEE Int. Conf. Commun.*, 2019-May, 2019. ISSN 15503607. doi: 10.1109/ICC.2019.8761300.

[30] Farah Haidar, Joseph Kamel, Ines Ben Jemaa, Arnaud Kaiser, Brigitte Lonc, and Pascal Urien. DARE: A Reports Dataset for Global Misbehavior Authority Evaluation in C-ITS. *IEEE Veh. Technol. Conf.*, 2020-May, 2020. ISSN 15502252. doi: 10.1109/VTC2020-Spring48590.2020.9128687.

[31] Xu Han, Jiawei Lu, Quan Yuan, and Jinglin Li. A Hierarchical Traffic-Balanced Route Planning Method for Connected Vehicles. *IEEE Veh. Technol. Conf.*, 2020-November, 2020. ISSN 15502252. doi: 10.1109/VTC2020-Fall49728.2020.9348792.

[32] Joseph Kamel, Mohammad Raashid Ansari, Jonathan Petit, Arnaud Kaiser, Ines Ben Jemaa, and Pascal Urien. Simulation Framework for Misbehavior Detection in Vehicular Networks. *IEEE Trans. Veh. Technol.*, 69(6):6631–6643, 2020. ISSN 19399359. doi: 10.1109/TVT.2020.2984878.

[33] Joseph Kamel, Michael Wolf, Rens W. Van Der Hei, Arnaud Kaiser, Pascal Urien, and Frank Kargl. VeReMi Extension: A Dataset for Comparable Evaluation of Misbehavior Detection in VANETs. *IEEE Int. Conf. Commun.*, 2020-June, 2020. ISSN 15503607. doi: 10.1109/ICC40277.2020.9149132.

[34] S B Kotsiantis and D Kanellopoulos. Data preprocessing for supervised leaning. *Int. J. ...*, 1(2):1–7, 2006. ISSN 1306-4428. doi: 10.1080/02331931003692557.

[35] Joyoung Lee and Byungkyu Park. Vehicle Intersection Control Algorithm Under the Connected Vehicles Environment. *IEEE Trans. Intell. Transp. Syst.*, 13(1):81–90, 2012.

[36] Ruiling Li, Fan Li, Xin Li, and Yu Wang. QGrid: Q-learning based routing protocol for vehicular ad hoc networks. *2014 IEEE 33rd Int. Perform. Comput. Commun. Conf. IPCCC 2014*, 2015. doi: 10.1109/PCCC.2014.7017079.

[37] Junwei Liang and Maode Ma. FS-MOEA: A Novel Feature Selection Algorithm for IDSs in Vehicular Networks. *IEEE Trans. Intell. Transp. Syst.*, pages 1–15, 2020. ISSN 1524-9050. doi: 10.1109/tits.2020.3011452.

[38] Junwei Liang and Maode Ma. An Efficiency-Accuracy Tradeoff for IDSs in VANETs with Markov-based Reputation Scheme. *IEEE Int. Conf. Commun.*, pages 1–6, 2021. ISSN 15503607. doi: 10.1109/ICC42927.2021.9500320.

[39] Xuejiao Liu, Oubo Ma, Wei Chen, Yingjie Xia, and Yuxuan Zhou. HDRS: A Hybrid Reputation System With Dynamic Update Interval for Detecting Malicious Vehicles in VANETs. *IEEE Trans. Intell. Transp. Syst.*, PP:1–12, 2021. ISSN 15580016. doi: 10.1109/TITS.2021.3117289.

[40] Jiawei Lu, Jinglin Li, Quan Yuan, and Bo Chen. A Multi-Vehicle Cooperative Routing Method Based on Evolutionary Game Theory. *2019 IEEE Intell. Transp. Syst. Conf. ITSC 2019*, pages 987–994, 2019. doi: 10.1109/ITSC.2019.8917441.

[41] Mohamed Nidhal Mejri, Nadjib Achir, and Mohamed Hamdi. A new security games based reaction algorithm against DOS attacks in VANETs. *2016 13th IEEE Annu. Consum. Commun. Netw. Conf. CCNC 2016*, pages 837–840, 2016. doi: 10.1109/CCNC.2016.7444896.

[42] Tarak Nandy, Rafidah Md Noor, Mohd Yamani Idna Bin Idris, and Sananda Bhattacharyya. T-BCIDS: Trust-Based Collaborative Intrusion Detection System for VANET. *2020 Natl. Conf. Emerg. Trends Sustain. Technol. Eng. Appl. NCETSTEA 2020*, 2020. doi: 10.1109/NCETSTEA48365.2020.9119934.

[43] NHTSA. Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey. Technical report, National Highway Traffic Safety Administration, 2015.

[44] NHTSA. Research Note 2018 Fatal Motor Vehicle Crashes : Overview. Technical report, National Highway Traffic Safety Administration, 2019.

[45] William S. Noble. What is a support vector machine? *Nat. Biotechnol.*, 24(12):1565–1567, 2006. ISSN 10870156. doi: 10.1038/nbt1206-1565.

[46] Melaouene Noussaiba and Romadi Rahal. State of the art: Vanets applications and their RFID-based systems. *2017 4th Int. Conf. Control. Decis. Inf. Technol. CoDIT 2017*, 2017-Janua:516–520, 2017. doi: 10.1109/CoDIT.2017.8102645.

[47] ORAD Committee. SAE J3016_202104. Technical report, SAE, 2021.

[48] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[49] L. E. Peterson. K-nearest neighbor. *Scholarpedia*, 4(2):1883, 2009. doi: 10.4249/scholarpedia.1883. revision #137311.

[50] M. Poongodi, Mounir Hamdi, Ashutosh Sharma, Maode Ma, and Pradeep Kumar Singh. DDoS Detection Mechanism Using Trust-Based Evaluation System in VANET. *IEEE Access*, 7:183532–183544, 2019. ISSN 21693536. doi: 10.1109/ACCESS.2019.2960367.

[51] Carlos H.O.O. Quevedo, Ana M.B.C. Quevedo, Gustavo A. Campos, Rafael L. Gomes, Joaquim Celestino, and Ahmed Serhrouchni. An Intelligent Mechanism for Sybil Attacks Detection in VANETs. *IEEE Int. Conf. Commun.*, 2020-June, 2020. ISSN 15503607. doi: 10.1109/ICC40277.2020.9149371.

[52] Walid Rjaibi, Sridhar Muppidi, and Mary O'Brien. Wielding a double-edged sword - Preparing cybersecurity now for a quantum world. Technical report, IBM, 2018. URL https://public.dhe.ibm.com/common/ssi/ecm/39/en/39017839usen/39017839usen-00{_}39017839USEN.pdf.

[53] Ayan Roy and Sanjay Kumar Madria. BLAME: A Blockchain-Assisted Misbehavior Detection and Event Validation in VANETs. *Proc. - IEEE Int. Conf. Mob. Data Manag.*, 2021-June(Mdm):69–78, 2021. ISSN 15516245. doi: 10.1109/MDM52706.2021.00021.

[54] SAE International. SAE J2735. Technical report, SAE, 2020.

[55] Ganis Zulfa Santoso and Moonsoo Kang. Performance analysis of AODV, DSDV and OLSR in a VANETs safety application scenario. *Int. Conf. Adv. Commun. Technol. ICACT*, pages 57–60, 2012. ISSN 17389445.

[56] Roshan Sedar, Charalampos Kalalas, and V Francisco. Reinforcement Learning-based Misbehaviour Detection in V2X Scenarios. *IEEE*, pages 2021–2023, 2021.

[57] Hichem Sedjelmaci and Sidi Mohammed Senouci. An accurate and efficient collaborative intrusion detection framework to secure vehicular networks. *Comput. Electr. Eng.*, 43: 33–47, 2015. ISSN 00457906. doi: 10.1016/j.compeleceng.2015.02.018. URL http://dx.doi.org/10.1016/j.compeleceng.2015.02.018.

[58] Hichem Sedjelmaci, Tarek Bouali, and Sidi Mohammed Senouci. Detection and prevention from misbehaving intruders in vehicular networks. *2014 IEEE Glob. Commun. Conf. GLOBECOM 2014*, pages 39–44, 2014. doi: 10.1109/GLOCOM.2014.7036781.

[59] Hichem Sedjelmaci, Imane Horiya Brahmi, Nirwan Ansari, and Mubashir Husain Rehmani. Cyber Security Framework for Vehicular Network Based on a Hierarchical Game. *IEEE Trans. Emerg. Top. Comput.*, 9(1):429–440, 2021. ISSN 21686750. doi: 10.1109/TETC.2018.2890476.

[60] Joan Serrà, David Álvarez, Vicenç Gómez, Olga Slizovskaia, José F. Núñez, and Jordi Luque. Input complexity and out-of-distribution detection with likelihood-based generative models. *ICLR 2020*, pages 1–15, 2019. URL http://arxiv.org/abs/1909.11480.

[61] Aekta Sharma and Arunita Jaekel. Machine Learning Approach for Detecting Location Spoofing in VANET. *Proc. - Int. Conf. Comput. Commun. Networks, ICCCN*, 2021-July:1–6, 2021. ISSN 10952055. doi: 10.1109/ICCCN52240.2021.9522170.

[62] Prinkle Sharma and Hong Liu. A Machine-Learning-Based Data-Centric Misbehavior Detection Model for Internet of Vehicles. *IEEE Internet Things J.*, 8(6):4991–4999, 2021. ISSN 23274662. doi: 10.1109/JIOT.2020.3035035.

[63] Steven So, Jonathan Petit, and David Starobinski. Physical layer plausibility checks for misbehavior detection in V2X networks. *WiSec 2019 - Proc. 2019 Conf. Secur. Priv. Wirel. Mob. Networks*, pages 84–93, 2019. doi: 10.1145/3317549.3323406.

[64] Steven So, Prinkle Sharma, and Jonathan Petit. Integrating Plausibility Checks and Machine Learning for Misbehavior Detection in VANET. *Proc. - 17th IEEE Int. Conf. Mach. Learn. Appl. ICMLA 2018*, pages 564–571, 2019. doi: 10.1109/ICMLA.2018.00091.

[65] Christoph Sommer, Reinhard German, and Falko Dressler. Bidirectionally coupled network and road simulation for improved IVC analysis. *IEEE Trans. Mob. Comput.*, 10(1):3–15, 2011. ISSN 15361233. doi: 10.1109/TMC.2010.133.

[66] Pavankumar Tallapragada and Jorge Cortes. Hierarchical-Distributed Optimized Coordination of Intersection Traffic. *IEEE Trans. Intell. Transp. Syst.*, 21(5):2100–2113, 2020. ISSN 15580016. doi: 10.1109/TITS.2019.2912881.

[67] Charles Tatkeu and Hauts De France. 2-Step Prediction for Detecting Attacker in Vehicle to Vehicle Communication. *IEEE*, pages 3–7, 2021.

[68] Lionel N Tidjone, Marc Frappier, and Amel Mammar. Intrusion Detection Systems : A Cross-Domain Overview. *IEEE*, 21(4), 2019.

[69] Awais Usman and Hamid Mukhtar. Design time considerations for cyber physical systems. *Proc. - 2012 IEEE Int. Conf. Green Comput. Commun. GreenCom 2012, Conf. Internet Things, iThings 2012 Conf. Cyber, Phys. Soc. Comput. CPSCom 2012*, pages 442–445, 2012. doi: 10.1109/GreenCom.2012.69.

[70] Rens W. van der Heijden, Thomas Lukaseder, and Frank Kargl. VeReMi: A dataset for comparable evaluation of misbehavior detection in VANETs. *Lect. Notes Inst. Comput. Sci. Soc. Telecommun. Eng. LNICST*, 254:318–337, 2018. ISSN 18678211. doi: 10.1007/978-3-030-01701-9_18.

[71] V. Vijayakumar, P. Inbavalli, K. Suresh Joseph, J. Amudhavel, D. Rajaguru, S. Sampath Kumar, T. Vengattaraman, and K. Premkumar. Quantitative analysis on various safety centric based approaches in VANET. *Glob. Conf. Commun. Technol. GCCT 2015*, pages 834–837, 2015. doi: 10.1109/GCCT.2015.7342778.

[72] P. Vinoth Kumar and M. Maheshwari. Prevention of Sybil attack and priority batch verification in VANETs. *2014 Int. Conf. Inf. Commun. Embed. Syst. ICICES 2014*, pages 0–4, 2015. doi: 10.1109/ICICES.2014.7033926.

[73] Jia Wu, Florent Perronnet, and Abdeljalil Abbas-Turki. Cooperative vehicle-actuator system: A sequencebased framework of cooperative intersections management. *IET Intell. Transp. Syst.*, 8(4):352–360, 2014. ISSN 1751956X. doi: 10.1049/iet-its.2013.0093.

[74] Jinqiao Wu, Min Fang, Haikun Li, and Xiao Li. RSU-Assisted Traffic-Aware Routing Based on Reinforcement Learning for Urban Vanets. *IEEE Access*, 8:5733–5748, 2020. ISSN 21693536. doi: 10.1109/ACCESS.2020.2963850.

[75] Chairit Wuthishuwong and Ansgar Traechtler. Vehicle to infrastructure based safe trajectory planning for autonomous intersection management. *2013 13th Int. Conf. ITS Telecommun. ITST 2013*, pages 175–180, 2013. doi: 10.1109/ITST.2013.6685541.

[76] Hao Yang, Hesham Rakha, and Mani Venkat Ala. Eco-Cooperative Adaptive Cruise Control at Signalized Intersections Considering Queue Effects. *IEEE Trans. Intell. Transp. Syst.*, 18(6):1575–1585, 2017. ISSN 15249050. doi: 10.1109/TITS.2016.2613740.

[77] Ti-Yen Yen and Wayne Wolf. Long Short Term Memory. *Hardware-Software Co-Synthesis Distrib. Embed. Syst.*, 1996. doi: 10.1007/978-1-4757-5388-2_2.

[78] Harry Zhang. Exploring conditions for the optimality of naïve bayes. *Int. J. Pattern Recognit. Artif. Intell.*, 19(2):183–198, 2005. ISSN 02180014. doi: 10.1142/S0218001405003983.

# Appendices

# Appendix A

# First Appendix

| Attack | SVM | KNN | DT | RF | VE |
|---|---|---|---|---|---|
| **1** | 6.489E-05 | 1.626E-03 | 1.022E-07 | 5.139E-06 | 1.705E-03 |
| **2** | 2.554E-04 | 1.539E-03 | 1.023E-07 | 1.074E-06 | 1.818E-03 |
| **3** | 4.717E-05 | 1.585E-03 | 1.279E-07 | 4.963E-06 | 1.639E-03 |
| **4** | 8.912E-05 | 2.430E-03 | 2.046E-07 | 5.601E-06 | 2.471E-03 |
| **5** | 6.612E-05 | 1.522E-03 | 7.670E-08 | 1.048E-06 | 1.594E-03 |
| **6** | 1.054E-04 | 1.524E-03 | 1.278E-07 | 4.525E-06 | 1.636E-03 |
| **7** | 5.996E-05 | 1.456E-03 | 7.673E-08 | 4.835E-06 | 1.526E-03 |
| **8** | 1.881E-04 | 1.453E-03 | 1.532E-07 | 4.526E-06 | 1.651E-03 |
| **9** | 4.992E-05 | 1.450E-03 | 7.632E-08 | 9.718E-07 | 1.502E-03 |
| **10** | 1.519E-04 | 1.553E-03 | 1.536E-07 | 4.915E-06 | 1.913E-03 |
| **11** | 4.358E-04 | 1.473E-03 | 2.046E-07 | 5.627E-06 | 1.916E-03 |
| **12** | 7.382E-05 | 1.506E-03 | 1.533E-07 | 5.057E-06 | 1.591E-03 |
| **13** | 1.041E-03 | 2.348E-03 | 2.553E-07 | 3.785E-05 | 3.491E-03 |
| **14** | 1.973E-05 | 1.695E-03 | 9.272E-08 | 4.822E-06 | 1.727E-03 |
| **15** | 2.932E-04 | 1.698E-03 | 1.391E-07 | 4.707E-06 | 2.025E-03 |
| **16** | 3.163E-04 | 1.769E-03 | 8.854E-08 | 9.519E-07 | 2.091E-03 |
| **17** | 4.156E-04 | 1.531E-03 | 1.789E-07 | 5.367E-06 | 1.961E-03 |
| **18** | 5.078E-05 | 1.619E-03 | 9.725E-08 | 3.598E-06 | 1.681E-03 |
| **19** | 2.675E-04 | 2.195E-03 | 1.946E-07 | 1.351E-06 | 2.443E-03 |
| **Average** | **2.101E-04** | **1.683E-03** | **1.391E-7** | **5.628E-06** | **1.912E-3** |

Table A.1: Full Model Execution Time Results by Attack Type