Cybersecurity in Connected and Automated Transportation Systems

by

Yiyang Wang

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Civil Engineering)
in the University of Michigan
2022

Doctoral Committee:

Associate Professor Neda Masoud, Chair
Professor Mingyan Liu
Professor Gábor Orosz
Professor Yafeng Yin

"*Learning without thinking leads to confusion; thinking without learning ends in danger.*"

– Confucius

Yiyang Wang
yiyangw@umich.edu
ORCID iD: 0000-0002-6478-7282

DEDICATION

To my mom, dad, and sister.

ACKNOWLEDGMENTS

It goes without saying that this work would have not come so far without the help of many people to whom I am indebted.

First, I would like to express my sincere gratitude to my advisor, Dr. Neda Masoud, for guiding me along this journey during the past four out of six years in University of Michigan. Dr. Masoud has not only been my academic mentor, but also a source of encouragement and affirmation when I stumbled and hesitated. Dr. Masoud always made time for me, and gave me plenty of freedom while allowing me to dive into interesting questions and conduct deep investigations, guided me with her profound insights and cared much about my personal development. I truly and greatly admire Dr. Masoud's passion towards research, devotion to teaching, and genuine care for her students. Thank you Dr. Masoud for being the best mentor I could ever wish for and I am so proud to be your Ph.D. student.

I would like to sincerely thank Dr. Mingyan Liu, Dr. Gábor Orosz, and Dr. Yafeng Yin for serving on my dissertation committee, and for providing invaluable insights and advice on shaping the dissertation. Additionally, I would also like to thank Dr. Xiuli Chao, and Dr. Tierra Bills for severing on my preliminary exam committee.

The work in this dissertation is a result of a joint collaboration with Dr. Anahita Khojandi, Dr. Henry Liu, and Ruixuan Zhang. I am greatly thankful for their contributions. Specifically, many thanks to Ruixuan for his hard work on the chapter on which we closely collaborated. I also had the opportunity to work with Dr. Robert Goodspeed and Meixin Yuan at University of Michigan, and Franco Van Wyk, Jeremy Watts, and Shahrbanoo Rezaei at University of Tennessee on projects that I am proud of, but are not part of this dissertation.

I am fortunate for having a group of dear friends during my stay in Ann Arbor in the past six years. I am grateful to Dr. Amirmahdi Tafreshian for his genuine and selfless help and advice for my academic endeavors during my Ph.D. studies. Also I would like to thank my friend Mojtaba Abdolmaleki for having countless, and always interesting, discussions with me. Many thanks to my friends Ethan Zhang, Benye Tang, Yicheng Zhou, Shihao Cheng, and Yuye Zhang for their company. I would like to thank all my friends at University of Michigan including but not limited to: Xiangguo Liu, Sahar Hemmati, Yuexi Tu, Huanyu Zhao, Roger Lloret, Zhen Yang, Xingmin Wang, Xintao Yan, Shuo Feng, Alex Sundt, Daniel Vignon, Tara Radvand, Xiaotong Sun, Zhengtian Xu, and Kidus Admassu. These friendships enriched my days in Ann Arbor.

Last but not least, I have been gifted with a kind and supporting family. Thanks to my twin

sister, Yiyuan, for her care and company. Yiyuan, I wish you all the best during your Ph.D. journey. In the end, I would like to thank my parents, Xin and Feng, for their unconditional and unlimited love during my Ph.D. studies.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

ABSTRACT

The last decade has been a witness to critical advancements in the field of intelligent transportation systems (ITS), with a great volume of research focusing on connected and automated vehicles (CAVs). A CAV leverages connected vehicle (CV) and automated vehicle (AV) technologies to enable automation and cooperation between vehicles and/or infrastructures to improve mobility, safety, efficiency, and sustainability in transportation systems. Despite these benefits, both connectivity and automation increase potential attack surfaces on CAVS, thereby introducing unprecedented cybersecurity challenges. This dissertation addresses these challenges by developing a comprehensive cybersecurity framework, which consists of two major components.

The first component of the framework focuses on anomaly detection in CAV sensors. A CAV requires high fidelity data for automated and cooperative driving tasks. However, oftentimes there exist uncertainties in the CAV sensor measurements, including noise and time delay. Therefore, the first part of the dissertation proposes a signal filtering-detection framework to estimate the CAV sensor state in the presence of potential cyberattacks. This dissertation considers sensor anomaly as a result of either malicious attacks or sensor faults, and proposes a series of data-driven detectors in conjunction with extended Kalman filter (EKF)-based algorithms for both signal filtering and anomaly detection. The first part of the dissertation consists of studies under different driving scenarios, including in car following and platooning modes. The study on the platooning mode further analyzes platoon stability under cybersecurity uncertainties, where a new class of string stability measures, namely pseudo string stability, is introduced to analyze the stability of a vehicle string under various types of attacks and imperfect detectors.

Despite the necessity of anomaly detection, a well-developed security monitor should also consider the dynamically changing environment faced by a CAV and the strategic behavior of the adversary to ensure both security and energy efficiency of the CAV. Therefore, the second component of the framework focuses on a macroscopic perspective to address security monitoring challenges faced by a CAV. First, the dissertation addresses the attack profile prediction and sensor selection problem under a security resource constraint. Toward this goal, the dissertation considers a sequential two-player game involving an attacker and a defender. This dissertation thereby develops an online learning algorithm for the defender to solve a variant of the multi-armed bandit (MAB) problem, namely, the multi-armed bandit with variable plays (MAB-VP). The dissertation provides a sublinear regret bound of the proposed algorithm, and derives the conditions under which a Nash equilibrium of the strategic game exists. Based on the analysis of the asymptotic expected reward of the two players, the dissertation assesses the effectiveness of the following

two defense strategies from a game theoretical perspective: (1) increasing security resources for monitoring, and (2) improving the performance of the detector.

Considering the fact that continuously monitoring all sensors aboard a CAV can be increasingly energy-intensive and therefore impractical, this dissertation introduces a dynamic security resource allocation problem to selectively monitor a subset of sensors for potential cyberattacks. This is accomplished by providing a mathematical framework based on a partially observable Markov decision process (POMDP), which prescribes a policy to dynamically assign security resource by balancing the trade-off between detection performance and energy-efficiency. This dynamic resource allocation problem serves as a complementary supplement to the sensor selection study, and together they form the last component of the proposed framework.

CHAPTER 1

Introduction

1.1 Background and Motivation

Our current transportation system is on the cusp of transforming into a highly connected, automated, and intelligent system (Ran & Boyce, 2012). The past decade has witnessed numerous advances in connected and automated vehicle (CAV) technology, which is considered to be an integral part of the future of intelligent transportation systems (ITS) (Shladover, 2018). CAVs have the potential to transform the ITS field by introducing numerous safety, mobility, and environmental sustainability benefits (Masoud & Jayakrishnan, 2017; van Wyk et al., 2019; Abdolmaleki et al., 2019, 2021). A CAV system combines connected vehicle (CV) and automated vehicle (AV) technologies, creating a synergistic impact that goes well beyond the benefits that each of these technologies can offer in isolation. It is envisioned that CAVs, with diverse degrees of connectivity and automation, will pave the path toward the next generation of transportation systems, which is more intelligent, efficient, and sustainable (Litman, 2017; Meyer & Beiker, 2019).

The synergy between CV and AV technologies originates from the additional data that the CV technology can provide to AV systems to improve their performance. The CV technology provides an integrated system of road users (e.g., vehicles, pedestrians, cyclists) and infrastructures whose performance can be jointly adjusted to satisfy a variety of social goals. Automation can surpass the human driver limitations, thereby greatly improving the transportation system performance. For instance, CAV technologies are expected to decrease fatal traffic crashes by as much as 80%, reducing their corresponding $870 billion cost, while also improving traffic flow and cutting into the approximate 7 billion hours American motorists spend in congested roads annually (USDOT, 2016). The CV communication links provide information beyond the range of sight of AV sensors, which is crucial for dampening shock waves in traffic (Van Arem et al., 2006). The CAV technology also extends and enhances currently available crash avoidance systems that use radars and cameras to detect collision threats by enabling CAVs to warn their surrounding vehicles of collisions and potentially hazardous circumstances (Bezzina & Sayer, 2014; Zhang et al., 2022).

CAVs provide mobility and sustainability benefits by enabling platoon formation, which can increase road capacity and traffic stability, and reduce fuel consumption (Van Arem et al., 2006; Ploeg et al., 2011; Liu et al., 2021, 2022). Platooning is enabled by using V2V or V2I technologies. It is found that the destabilization effect of communication delay is suppressed by the stabilization effect of multi-anticipations of platoons (Ngoduy, 2015). Platooning can also reduce fuel consumption, which is significantly influenced by air resistance, through shorter following gaps. It has been showed that tightly coupled platooning can potentially improve fuel economy for both large trucks (Alam, 2011; Sun, 2020) and passenger vehicles (Shida & Nemoto, 2009). One application of platooning is cooperative adaptive cruise control (CACC), which is a further development of adaptive cruise control (ACC) that adds V2V communication, providing the ACC system with more and higher-quality information about the vehicle(s) it is following. With information of this type, the ACC controller will be able to better anticipate the challenges ahead, enabling it to be safer, smoother, and provide a more "natural" response. CACC systems utilizing V2V communication could allow the mean following time gap to be reduced from about 1.4 s in manual driving to approximately 0.6 s (Nowakowski et al., 2010).

Although CAVs offer promising benefits, they are prone to various security and privacy risks. In particular, the security risk escalates with increasing levels of connectivity and automation. CAVs use a variety of sensors to build a virtual map of their surrounding environment in order to drive in the correct lane within the speed limit, avoid collisions, and detect obstacles in their immediate physical environment. Meanwhile, the interconnection between the infrastructures and vehicles also relies on various types of sensors to provide state information and situational awareness. Hence, anomalous information due to either malicious cyberattacks or faulty vehicle sensors can pose safety risks to road users, and possibly cause fatal crashes. For instance, recently there have been demonstrated cyberattacks on vehicle sensors in (Cao et al., 2019a,b), where the authors use optimization-based approaches to fool the light detection and Ranging (LiDAR) sensors aboard vehicle. At the system level, the infrastructures and vehicles can be viewed as individual nodes in a large interconnected network, where a single malicious attack on a subset of sensors of one node can easily propagate through this network, affecting other network components (e.g., other vehicles, traffic control devices, etc.). For example, Feng et al. (2018) demonstrated that by sending falsified data to actuated and adaptive signal control systems, a malicious hacker could increase total system delay in a real-world corridor. Therefore, there is an increasing need for cyber security solutions, especially for sensor security solutions, to enhance the safety and reliability of the entire system.

Prevention is normally recognized as one of the best defense strategies against malicious hackers or attackers. Predicting attack profiles enables the system to deploy better prevention mechanisms, where behaviors of both the attacker and the defender have to be considered. Although there is a large body of literature addressing sensor security in ITS (Van Wyk et al., 2019; Wang et al., 2020c;

Yang & Lv, 2021; Alotibi & Abdelhakim, 2020; Watts et al., 2022), most of the existing work mainly focuses on sensor intrusion/anomaly detection without attack profile analysis. A security monitor is therefore needed to predict the attack strategy of the attacker so that it can be quickly identified and mitigated, before detecting the attack on the subject CAV.

At the same time, there is generally a trade-off between the desired outcomes (e.g., the accuracy of system state estimation) and the cost or energy associated with cyberattack detection and prevention (e.g., scanning subsets of sensors, deploying redundant sensor systems, etc.), as oftentimes there exists limited energy for system operations. For example, previous studies have reported limited energy availability in CAVs, wireless sensor networks, and Internet of Things (IoT) networks (Cheng et al., 2005; Arroyo-Valles et al., 2007; Kim & Jung, 2019). There also exists limited bandwidth constraint in networks such as body sensor networks or IoT sensors and devices (Nabar et al., 2010; Guegan & Orgerie, 2019; Saghafian et al., 2022). As more sensors are installed on CAVs, and given the cost, computational burden, and bandwidth and energy constraints associated with security monitoring, a dynamic security resource allocation scheme is needed to ensure both security and energy efficiency in CAVs. Such a decision-making problem needs to consider various information including the system states, potential security threats, and the surrounding environment. Moreover, in practice not all the aforementioned information can be directly and completely quantified, and can only be estimated through partial observations.

In the next section, we introduce several characteristics of CAV systems that pose a challenge in developing cybersecurity solutions.

1.2 Challenges

- **State Estimation under Cybersecurity Uncertainties:** To ensure that a CAV can safely and effectively navigate the network, it needs access to robust and accurate data streams. As a result, any anomalous sensor data if undetected, can greatly imperil the decision making process of CAVs. The presence of an anomaly in the data collected from CAV sensors can imply (i) a subset of sensors are faulty, or (ii) there has been a malicious cyberattack. Moreover, with the presence of measurement noise, it is crucial to detect any anomalous sensor readings in real-time and accurately recover the true state of the CAV to ensure its safety.
- **Time Delay:** Connectivity enables a CAV to receive information from its surrounding vehicles and infrastructures. However, wireless communications and some forms of cyberattacks (e.g., the jamming attack) can introduce time delays to the receiver, which can significantly impact the state estimation accuracy and traffic stability, causing delay and chaos and even fatal crashes. Therefore, it is imperative to take time delay into consideration during state estimation

and anomaly detection processes, and investigate the potential impacts of time delay on traffic stability.

- **Unknown Behavior of the Adversary:** In order to deploy better prevention mechanisms, behaviors of both the attacker and the defender have to be considered so that the attack profile can be predicted. However, in practice the attacker's behaviour is unknown most of the time. A prediction of the adversary's behavior is needed, while this prediction needs to account for the strategic decision making by and interactions between the attacker and the defender.
- **Constrained Security Resources:** As more sensors are mounted aboard CAVs or installed on the transportation infrastructure, it becomes more difficult to monitor the sensors continuously for potential cyberattack detection, mainly due to the limited resources available to the CAV. Thus, it is utterly important to consider dynamic allocation of security resources for threat monitoring.

1.3 Dissertation Outline

This dissertation aims to investigate the cybersecurity challenges faced by CAVs and to provide a holistic defense framework. In what follows, we provide a brief overview of the chapters, which collectively reveal the contributions of this dissertation.

In Chapters 2-4, we develop a novel observer-based method to improve the safety and security of CAV transportation. The proposed method combines model-based signal filtering and anomaly detection methods. Specifically, in Chapter 2, we develop an adaptive extended Kalman filter (AEKF) to smooth sensor readings of a CAV based on a nonlinear car-following model. Using the car-following model the subject vehicle (i.e., the following vehicle) utilizes the leading vehicle's information to detect sensor anomalies by employing previously-trained One Class Support Vector Machine (OCSVM) models. This approach allows the AEKF to estimate the state of a vehicle not only based on the vehicle's location and speed, but also by taking into account the state of the surrounding traffic. A communication time delay factor is considered in the car-following model to make it more suitable for real-world applications. In Chapter 3, we devise an augmented state formulation of extended Kalman filter, namely the augmented state extended Kalman filter (ASEKF). The augmented state formulation enables the Kalman filter to compensate for potential biases caused by model inaccuracy and time delay, thereby improving the detection performance. To make the proposed model more suitable for real-world applications, we consider a stochastic communication time delay in the car-following model. In Chapters 2 and 3 we only utilize one leading vehicle's information to detect sensor anomalies. However, a CAV can receive and utilize information from multiple sources for planning purposes. In Chapter 4, we extend the car following setting in the previous chapters to a more general platooning setting, where a CAV can receive

information from multiple vehicles in the platoon. We develop a comprehensive framework to model the impact of cyberattacks on safety, security, and head-to-tail stability of connected and automated vehicular platoons. First, we propose a general platoon dynamics model with heterogeneous time delays that may originate from the communication channel and/or vehicle onboard sensors. Based on the proposed dynamics model, we adopt ASEKF to smooth the sensor reading, and use it in conjunction with an anomaly detector to detect sensor anomalies. Furthermore, we introduce a novel concept in string stability, namely, pseudo string stability, to measure a platoon's string stability under cyberattacks and model uncertainties. We demonstrate the relationship between pseudo string stability of a platoon and its detection recall/sensitivity, which enables us to identify the critical detection sensitivity/recall that the platoon's members should meet for the platoon to remain pseudo string stable.

In Chapters 2-4, we model a CAV's motion and detect sensor anomaly at the microscopic level, where we investigate the aforementioned challenges including state estimation under cybersecurity uncertainties and time delay. However, we have not yet tackled the challenges associated with the unknown behavior attacker and constrained security resources. In Chapter 5, we focus on these challenges by modelling the attacker and defender behavior as a multi-armed bandit (MAB) problem. Specifically, we extend the adversarial/non-stochastic multi-play multi-armed bandit (MPMAB) to the case where the number of arms to play is variable. The work is motivated by the fact that the resources allocated to scan different critical nodes in an interconnected transportation network change dynamically over time and depending on the environment. By modeling the malicious hacker and the intrusion monitoring system as the attacker and the defender, respectively, we formulate the problem for the two players as a sequential pursuit-evasion game. We derive the condition under which a Nash equilibrium of the strategic game exists. For the defender side, we provide an exponential-weighted based algorithm with sublinear pseudo-regret. We further extend our model to heterogeneous rewards for both players, and obtain lower and upper bounds on the average reward for the attacker. We provide numerical experiments to demonstrate the effectiveness of a variable-arm play.

Although we have considered constrained security resources in Chapter 5, the MPMAB problem does not dynamically determine the amount of security resource for monitoring. Therefore, in Chapter 6, we aims to fully address this last challenge, i.e., dynamic security resource allocation for a CAV. We consider an imperfect detector that allows us to only partially observe whether the CAV is under attack, and formulate the problem as a partially observable Markov decision process (POMDP) model, which prescribes a dynamic security resource allocation policy to minimize the total expected discounted cost of a trip to ensure both security and energy efficiency of the CAV. We demonstrate the effectiveness of our proposed model in the numerical experiments. Finally, Chapter 7 concludes this dissertation and further discusses possible directions for future work.

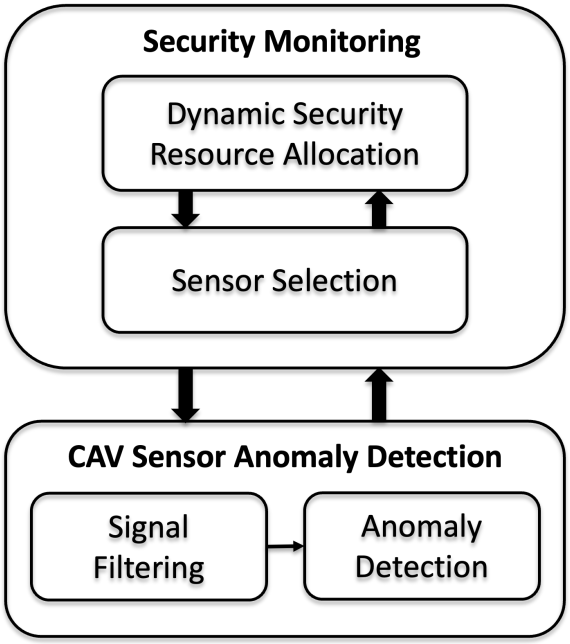


Figure 1.1: The hierarchical defense framework of CAV system.

Figure 1.1 illustrates the hierarchical defense framework proposed in this dissertation. The framework consists of two parts: First, Chapters 2-4 provides solutions for CAV sensor anomaly detection; Secondly, Chapters 5-6 address the security monitoring task, including dynamic security resource allocation and sensor selection. The two parts work together to offer a comprehensive framework, where the security monitor determines the allocation of security resources and the subset of sensors to be scanned, and the detection results provide feedback to the security monitor for future decision-making.

The main chapters of this dissertation, Chapters 2 through 6, have appeared in the following five publications, respectively:

- Wang, Y., Masoud, N., & Khojandi, A. (2020c). Real-time sensor anomaly detection and recovery in connected automated vehicle sensors. *IEEE transactions on intelligent transportation systems*, 22(3), 1411–1421
- Wang, Y., Masoud, N., & Khojandi, A. (2020b). Anomaly detection in connected and automated vehicles using an augmented state formulation. *2020 Forum on Integrated and Sustainable Transportation Systems (FISTS)*, 156–161
- Wang, Y., Zhang, R., Masoud, N., & Liu, H. X. (N.D.). Anomaly detection and string stability analysis in connected automated vehicular platoons. *Under first round of review*
- Wang, Y. & Masoud, N. (2021). Adversarial online learning with variable plays in the pursuit-evasion game: Theoretical foundations and application in connected and automated vehicle cybersecurity. *IEEE Access*, 9, 142475–142488

- Wang, Y. & Masoud, N. (N.D.). Dynamic security resource allocation for connected and automated vehicles. *To be submitted*

1.4 Contribution

This dissertation provides a comprehensive defense framework to address the cybersecurity challenges faced by CAVs. To this point, the contributions of this dissertation can be summarized as follows:

- There exists a gap in the literature of state estimation in CAVs in the presence of potential cyberattacks. This dissertation is the first to propose a filtering-detection framework to fill this gap. Specifically, information from multiple vehicles (the leading vehicle in the car following mode and multiple vehicles in the platooning mode) are utilized for both filtering and attack detection while accounting for potential time delays introduced by the communication channel and/or sensor data collection/processing.

- This dissertation is the first to conduct stability analysis on CAV platoons under cybersecurity uncertainties. Specifically, this dissertation puts forward a new class of string stability measures, namely, pseudo string stability, to analyze the stability of a vehicle string under cyberattacks. Pseudo string stability analysis enables us to obtain a critical detection recall/sensitivity that ensures a pseudo stable string, under different attack settings.

- This dissertation is the first to model the pursuit-evasion game played between an attacker and a defender (i.e., the security monitor in a CAV) using the bandit problem with variable plays. The defender's decisions is modeled as a multi-armed bandit problem with variable plays. The current literature of bandit problem does not consider a variable-plays setting, and therefore cannot be directly applied in scenarios with dynamically changing resource constraints. This dissertation develops a no-regret algorithm to adaptively learn the attacker's behaviour and subsequently select a subset of sensors to scan under limited and time-dependent security resources. Theoretical guarantees are provided on the performance of the proposed algorithm. This is also the first study addressing the variable-plays setting from a game theoretic perspective in the online learning literature. The developed algorithms and theoretical results can be applied beyond the specific application of cybersecurity in CAVs.

- As it is not practical to continuously monitor all sensors on a CAV, the security resources need to be dynamically allocated throughout a trip. However, the dynamic security resource allocation it not yet widely studied in the literature. This dissertation addresses this gap

by developing a POMDP model to dynamically assign security resources to ensure energy efficiency.

CHAPTER 2

Real-time Sensor Anomaly Detection and Recovery in Connected Automated Vehicle Sensors

2.1 Introduction

Anomaly detection in CAV sensors is an important but also challenging task. A traveling CAV could use the most recent history of data to detect anomalies. Presence of an anomaly in the pattern of data collected from a CAV sensor system can imply (i) a subset of sensors are faulty, or (ii) there has been a malicious attack. In both cases, it is vital to detect the anomalies and exclude the anomalous data from the decision making process.

An anomaly detection scheme introduces two types of errors – false negatives and false positives. It is easy to see that a false negative error can allow falsified data to affect trajectory planning, which could lead to fatal consequences. Although less apparent, a false positive error can have consequences that are just as severe. Consider a situation where an actual event in the network (e.g., an unexpected braking from a downstream vehicle) has led to an abrupt change in the pattern of observed data. If the vehicle falsely detects such an unexpected change as a fault/attack and discards the information, it may lead to the CAV not reacting to such abrupt changes in the network appropriately and in a timely manner, creating dangerous, and potentially fatal, scenarios. In order to prevent this type of false positive error, it is necessary for vehicles to incorporate network-level information in their anomaly detection scheme.

In addition to distinguishing between real changes in network conditions and anomalies, the anomaly detection methods should be able to identify the noise introduced by sensors and the communication channel, exacerbated by potential communication delay, as well as the missing values in the collected data. Moreover, due to resource constraints for each vehicle, the anomaly detection techniques in CAVs need to be lightweight, and implementable in real-time.

Anomalous sensor behavior could manifest itself in various forms and representations. Several faulty sensor behaviors are discussed in (Sharma et al., 2010). Petit & Shladover (2015) summarize the taxonomy of intrusions or attacks on automated vehicles, among which the false injection attack

is considered to be the most dangerous attack. In this paper, we consider five types of the anomalous sensor behavior resulting from both sensor faults and false injection attacks. We base this paper on the sensor failure and/or attack taxonomy provided by Sharma et al. (2010) and Van Wyk et al. (2019):

1. Short: A single, sharp and abrupt change in the observed data between two successive sensor readings.
2. Noise: An increase in the variance of the sensor readings. Different from Short, the Noise anomaly type occurs across *multiple* successive sensor readings.
3. Bias: A temporarily constant offset from the sensor readings.
4. Gradual drift: A small and gradual drift in observed data during a time period. Over time, a gradual drift can result in a large discrepancy between the observed and the true state of the system.
5. Miss: Lack of available data during a time period.

We do not explicitly account for 'miss,' which can result from DoS attacks preventing the exchange of information. However, note that 'miss,' depending on its duration, can be viewed as 'short' or 'bias', where the sensor reading is non-existent instead of showing a wrong value. Hence, it can partially be addressed using the same methods for detecting 'short' or 'bias'. For examples of specific scenarios that could lead to these anomalies, refer to (Ni et al., 2009).

In order to successfully detect different types of anomalies, avoid falsely identifying unexpected changes in the network as anomalies, and mitigate the impact of random noise/missing values, we develop a novel and comprehensive framework that combines the adaptive extended Kalman Filter (AEKF) with a car following motion model, and employs a data-driven fault detector. Our framework is capable of accounting for delay in observing the environment, introduced by a congested communication channel and/or delayed sensor observation. Specifically, we use a car-following model to govern the motion of the vehicle in order to capture the interaction between the subject vehicle and its immediate leading vehicle. We demonstrate that the time delay incorporated in the motion model renders the traditional χ^2 fault detector (Brumback & Srinath, 1987) not appropriate for anomaly detection, and propose and implement a One Class Support Vector Machine (OCSVM) model for anomaly detection (Schölkopf et al., 2001) together with AEKF instead. We demonstrate the power of the proposed framework in detecting various types of anomalies.

Our main objective in this study is to detect sensor anomalies and to recover the corrupt signals by utilizing the surrounding vehicles' information. To this end, the following assumptions are made:

1. Vehicles move according to a car-following model (i.e., under adaptive cruise control mode), have access to location and velocity of their leader (either through BSMs or using their on-board sensors), and are able to control over their own acceleration rates.
2. A known time delay (e.g., communication, sensing, and/or reaction delay) is applied to the

input vector of the car-following model.

The rest of the chapter is organized as follows: Section 2.2 provides a brief review of the existing related work in the field of anomaly detection in CAVs. Section 2.3 introduces the formulation of the problem and our method. In Section 2.4 we conduct a case study based on a well-known car-following model. Finally, in Section 2.5, we conclude this paper.

2.2 Literature Review

Anomaly detection research has generated a substantial volume of literature over the past few years, as it is an important and challenging problem in many disciplines including but not limited to automotive systems (Müter & Asaj, 2011; Müter et al., 2010), wireless networks (Rajasegarar et al., 2008), and environmental engineering (Hill et al., 2007; Hill & Minsker, 2010). Anomaly detection methods are used in a variety of applications including fault diagnosis, intrusion detection, and monitoring applications. In some cases if the source of an anomaly can be quickly identified, appropriate reconfiguration control actions can be made in order to avoid or minimize potential loss.

In the past few years, a variety of methods have been developed to detect anomalous behavior, and/or identify the source of anomaly (Isermann, 1984; Hwang et al., 2010). Examples of anomaly detection methods include observer-based methods (Clark et al., 1975; Wünnenberg & Frank, 1987), parity relation methods (Deckert et al., 1977; Gertler, 1997), and parameter estimation methods (Baskiotis et al., 1979), etc. Among them, observer-based (quantitative model-based) fault detection is a common fault detection approach, as discussed in (Hwang et al., 2010). Observer-based fault detection is based on the residual (or innovation) sequence obtained from using a mathematical model and (adaptive) thresholding. In this chapter we study anomaly detection in CAVs using observer-based anomaly detection.

Anomalous sensor behavior in CAVs could result from both sensor failures or malicious cyberattacks. Sensor readings may be influenced by a variety of factors, leading to collection of faulty information (Checkoway et al., 2011; Realpe et al., 2015; Pous et al., 2017). For example, environmental perturbations and sensor age may result in higher probabilities of failure. A short circuit, loose wire connection, or low battery supply are among other reasons that may cause inaccurate data reporting, including an unexpectedly high variation of sensor reading, or noise (Ni et al., 2009).

Additionally, malicious attacks may cause anomaly in sensor readings. CAVs have several internal and external cyberattack surfaces through which they can be accessed and compromised by ill-intended actors (Petit & Shladover, 2015; Checkoway et al., 2011; Koscher et al., 2010; Weimerskirch & Gaynier, 2015; Yan et al., 2016). Petit & Shladover (2015) showed that false injection of information and map database poisoning are two of the most dangerous potential attacks

on CAVs. For example, the infrastructure (i.e., RSU) or a neighboring vehicles can transmit fake messages (e.g., WAVE Service Advertisement, BSM), which may in turn generate wrong, and potentially harmful, reactions (e.g., spurious braking), placing CAV occupants and other road users in life-threatening situations. There are several existing studies that illustrate the vulnerability of CAV sensors, e.g., speed, acceleration and location sensors, to cyberattacks or sensor faults. For in-vehicle speed and acceleration sensors, a false injection attack mentioned in (Petit & Shladover, 2015) through the CAN bus or the on-board diagnostics (OBD) system could induce any of the four types of anomalies considered in this paper. As another example, Trippel et al. (2017) demonstrated that an acoustic injection attack could lead to anomalous sensor values for the in-vehicle acceleration sensor. Lastly, for the location measurement from the GPS, both the operating environment of the vehicle and GPS spoofing/jamming attacks may result in anomalous sensor values (Faughnan et al., 2013). Note that in this chapter we only consider false injection attacks whose manifestations can be described by four types of anomaly defined in Section I. As such, the chapter leaves out any types of attacks that do not impact sensor readings.

Despite the severe consequences of failing to detect sensor anomalies in CAVs, there is a scarcity of anomaly detection techniques in the ITS literature. Only a limited number of studies have focused on cyber security in CAVs, or more generally in ITS. In (Park et al., 2015), the authors used graph theory based on a transient fault model to detect transient faults in CAVs. Christiansen et al. (2016) combined background subtraction and convolutional neural networks to detect anomalies/obstacles. Müter & Asaj (2011) and Marchetti et al. (2016) used entropy-based methods to detect anomalies (attacks) in in-vehicle networks. Faughnan et al. (2013) measured the discrepancy between redundant sensor readings to detect hijacking in unmanned aerial vehicles. Van Wyk et al. (2019) used a CNN - Kalman Filter - χ^2-detector hybrid method to detect and identify sensor anomaly in a CAV system.

In this chapter, we focus on detection of anomalous sensor readings and recovery of the corrupt signals. We propose an observer-based anomaly detection method, which combines a well-known filtering technique, namely, AEKF, to smooth the CAV sensor values, and a machine learning method, i.e., OCSVM, to learn the normal vehicle behavior, with the objective of detecting anomalous behavior. Specifically, we utilize a car following model to take into account the information from the leading vehicle, so as to better detect anomalies by reducing the false positive error rate. Additionally, to make our methodology robust to practical network conditions and improve its anomaly detection performance, we account for time delay in perceiving the environment, which could arise from communication delay or sensor observation delay.

One of the major differences of this chapter with our past work is that in (Van Wyk et al., 2019) we examine multiple sensor readings for each type of sensor at the same time by feeding multiple sensor readings into a CNN network. However, in this chapter, for each type of sensor we rely on readings from a single sensor only and propose a novel anomaly filtering and detection technique

accordingly. Another major difference is that this work takes into account the state of the leading vehicle when conducting anomaly detection for the subject vehicle. These two major differences make the two frameworks fundamentally different, and applicable to different scenarios. Finally, in this chapter we have replaced the traditional χ^2-detector, which was used in (Van Wyk et al., 2019), with a OCSVM model. Our experiments show that by cooperating leading vehicle's information and using OCSVM, we achieve a better detection performance compared to the traditional χ^2-detector. To the best of our knowledge, this is the first study that detects CAV sensor anomaly by utilizing leading vehicle's information, i.e., by incorporating a car-following model into a continuous state-space model with time delay. Additionally, given the fact (demonstrated in the paper) that the model noise does not follow a Gaussian distribution, rendering the traditional χ^2 test inapplicable, we propose an OCSVM model in order to deal with the bias and abnormal distribution of innovation caused by time delay.

2.3 Solution Methodology

In this section, we first discuss how a car-following model with time delay can be used to describe the motion (also known as state-transition) model in AEKF. Next, we formulate a new continuous nonlinear state-space model with discrete measurement based on a car-following motion model. The continuous state-transition model represents the intrinsic nature of a vehicle's response to the actions of its immediate downstream traffic, and the discrete measurement model represents the mechanics of sensor sampling, as is the case in practice. Based on the proposed state-space model, we propose an anomaly detection method, which combines AEKF and OCSVM. Also a traditional χ^2-detector is discussed and its performance is compared with that of OCSVM.

2.3.1 Car-Following Model with Time Delay

Consider the car-following model in (Treiber & Kesting, 2012):

$$\begin{aligned} d_n(t) &= x_{n-1}(t) - x_n(t) \\ \dot{d}_n(t) &= v_{n-1}(t) - v_n(t) \\ \dot{v}_n(t) &= f(v_n(t-\tau), d_n(t-\tau), \dot{d}_n(t-\tau)) \end{aligned} \tag{2.1}$$

where $\dot{v}_n(t)$, $v_n(t)$, $x_n(t)$ are respectively the acceleration, speed, and location of the nth vehicle, to which we refer as the 'subject vehicle', and $d_n(t)$ and $\dot{d}_n(t)$ are the distance gap and the speed difference between the subject vehicle and its leading vehicle, the $(n-1)$th vehicle, respectively. Parameter τ denotes time delay, also known as the 'perception-reaction time', i.e., the period of

time lapsed from the moment the leading vehicle performs an action, to the moment the subject vehicle executes an action in response. Function f is the stimulus function.

$\dot{v}_n(t)$ in Equation (2.1) can be recast in the following form:

$$\dot{v}_n(t) = q\left(x_n(t-\tau), v_n(t-\tau), x_{n-1}(t-\tau), v_{n-1}(t-\tau)\right) \tag{2.2}$$

where q produces the same output as f, given a different set of inputs.

We define a state vector in continuous time as:

$$s_n(t) = [x_n(t), v_n(t)]^\intercal \in \mathbb{R}^2 \tag{2.3}$$

where $x_n(k) \in \mathbb{R}$ and $v_n(k) \in \mathbb{R}$. Note that, without loss of generality, x and v can be extended to vector form to allow for incorporating historical location and speed observations, respectively, into the state-space model, when desired.

Recasting equation (2.2) as a function of $s_n(t)$ produces a car following model that maps the state into an actionable decision for the subject vehicle:

$$\dot{v}_n(t) := f_v\left(s_n(t-\tau), u_n(t-\tau)\right) \tag{2.4}$$

where $u_n(t) = [x_{n-1}(t), v_{n-1}(t)]^\intercal$ is the input vector containing information received from the leading vehicle, and f_v denotes the stimulus function describing velocity in a continuous state space.

2.3.2 Continuous State-Discrete Measurement State-Space Model

We now define a state-space model with a continuous state-transition model and discrete measurements. Using previous definition of the state vector $s_n(t)$, the state-transition model satisfies the following differential equation:

$$\begin{aligned} \dot{s}_n(t) &= \begin{bmatrix} \dot{x}_n(t) \\ \dot{v}_n(t) \end{bmatrix} \\ &= \begin{bmatrix} e_2^\intercal s_n(t) \\ f_v(s_n(t-\tau), u_n(t-\tau)) \end{bmatrix} \end{aligned} \tag{2.5}$$

where $e_2 = [0, 1]^\intercal$.

When $\tau = 0$, the state-space model in equation (2.5) satisfies the Markovian property, allowing for applying AEKF. However, in practice, a variety of factors including time required for data processing and computations as well as delays in the communication network can cause τ to be non-zero. As such, in practice AEKF cannot be applied to equation (2.5), since the derivative of the

state vector is determined by multiple previous state vectors.

In order to apply AEKF, we approximate equation (2.5) in the following way: We assume the acceleration of each vehicle is bounded within the interval $[a_{\min}, a_{\max}]$, where $a_{\min} \leq 0$ and $a_{\max} > 0$ indicate the magnitude of the maximum deceleration and acceleration rates, respectively. Based on the assumption of bounded acceleration, we can obtain lower and upper bounds on the approximation of $v_n(t)$:

$$e_2^\intercal s_n(t-\tau) + a_{\min}\tau \leq v_n(t) \leq e_2^\intercal s_n(t-\tau) + a_{\max}\tau$$

Then a delay differential equation (DDE), describing the delayed state-transition model, can be used to approximate equation (2.5):

$$\begin{aligned} \dot{s}_n(t) &= \begin{bmatrix} \dot{x}_n(t) \\ \dot{v}_n(t) \end{bmatrix} \\ &= \begin{bmatrix} e_2^\intercal s_n(t-\tau) + \int_{t-\tau}^{t} a_n(r)dr \\ f_v(s_n(t-\tau), u_n(t-\tau)) \end{bmatrix} \\ &\approx \begin{bmatrix} e_2^\intercal s_n(t-\tau) \\ f_v(s_n(t-\tau), u_n(t-\tau)) \end{bmatrix} \\ &= \mathcal{T}(s_n(t-\tau), u_n(t-\tau)) \end{aligned} \tag{2.6}$$

where $a_n(t)$ is the acceleration of the nth vehicle at time t, and $\mathcal{T}$ denotes the state-transition model. Finally, we obtain a continuous-time state-transition model with discrete-time measurement as the following:

$$\begin{aligned} \dot{s}_n(t) &= \mathcal{T}(s_n(t-\tau), u_n(t-\tau)) + \theta(t) \\ z_n(t_k) &= \mathcal{M}(s_n(t_k)) + \eta(t_k),\ k \in \{0 \cup \mathbb{Z}^+\} \end{aligned} \tag{2.7}$$

where $\mathcal{M}(\cdot)$ is the measurement model, $z_n(t_k)$ denotes sensor reading of the nth vehicle, $\theta(t)$ and $\eta(t_k)$ are the process noise and the observation noise, respectively, which are assumed to be mutually independent, $t_{k+1} = t_k + \Delta t,\ k \in \{0 \cup \mathbb{Z}^+\}$, and Δt is the sampling time interval for sensors. Note that $\theta(t)$ accounts for the error introduced by the approximation steps in equation (2.6).

2.3.3 Adaptive Extended Kalman Filter with Fault Detector

Extended Kalman Filter (EKF) is a well-established method used for timely and accurate estimation of the dynamic state of a non-linear system (Wan, 2006). One important issue that needs to be addressed in EKF is how to properly set up the covariance matrices of process noise (i.e., Q) and

measurement noise (i.e., R). The performance of EKF is highly affected by proper tuning of Q and R (Mohamed & Schwarz, 1999), while in practice these parameters are usually unknown a priori. Therefore, we apply an adaptive extended Kalman filter (AEKF) to approximate these matrices.

An EKF is used to estimate state vector $s_n(t_k)$ from sensor reading $z_n(t_k)$. Let $\hat{s}(k|k-1)$ and $P(t_k|t_{k-1})$ denote the state prediction and state covariance prediction at time t_k, given the estimate at time t_{k-1}, respectively. Note that for ease of notation, we omit subscript n. Hence, considering the state-space model in equation (2.7), the EKF consists of the following 3 steps:

Step 0 - Initialize state mean and covariance:

$$\begin{aligned} \hat{s}_{k-1|k-1} &= \mathbb{E}[s(t_0)] \\ P_{k-1|k-1} &= \text{Var}[s(t_0)] \end{aligned} \tag{2.8}$$

Step 1 - Predict state and state covariance:

$$\begin{aligned} &\text{Solve} \begin{cases} \dot{\hat{s}}(t) = \mathcal{T}(\hat{s}(t-\tau), u(t-\tau)), \\ \dot{P}(t) = F(t-\tau)P(t-\tau) + P(t-\tau)F(t-\tau)^\intercal + Q(t) \end{cases} \\ &\text{with} \begin{cases} \hat{s}(t_{k-1}) = \hat{s}_{k-1|k-1} \\ P(t_{k-1}) = P_{k-1|k-1} \\ t_{k-1} \geq \tau + t_0 \end{cases} \\ &\Rightarrow \begin{cases} \hat{s}_{k|k-1} = \hat{s}(t_k) \\ P_{k|k-1} = P(t_k) \end{cases} \end{aligned} \tag{2.9}$$

where $F(t-\tau) = \frac{\partial \mathcal{T}}{\partial s}|_{\hat{s}(t-\tau), u(t-\tau)}$ is the first-order approximation of the Jacobian matrix of the state-transition model $\mathcal{T}(\cdot)$.

Step 2 - Update state and state covariance:

$$\begin{aligned} \nu_k &= z(t_k) - \mathcal{M}(\hat{s}_{k|k-1}) \\ S_k &= H(t_k)P_{k|k-1}H(t_k)^\intercal + R_k \\ K_k &= P_{k|k-1}H(t_k)^\intercal S_k^{-1} \\ \hat{s}_{k|k} &= \hat{s}_{k|k-1} + K_k\nu_k \\ P_{k|k} &= P_{k|k-1} - K_kH(t_k)P_{k|k-1} \end{aligned} \tag{2.10}$$

where $H(t_k) = \frac{\partial \mathcal{M}}{\partial s}|_{\hat{s}_{k|k-1}}$, $Q(t)$ is the covariance matrix of the process noise at time t, $R_k = R(t_k)$ is the covariance matrix of the measurement noise at time t_k, and ν_k is innovation, which is the difference between the measurement and the prediction at time t_k.

Since in practice Q and R are usually unknown, based on the work in (Akhlaghi et al., 2017) with slight modifications, we apply an AEKF to estimate these two matrices by using a moving estimation window of size M, as follows:

$$\begin{aligned}
\mu_k &= z(t_k) - \mathcal{M}(\hat{s}_{k|k}) \\
\hat{R}_k &= \sum_{i=1}^{M} \lambda_i \left(\mu_{k-i+1}\, \mu_{k-i+1}^{\intercal} + H(t_{k-i+1}) P_{k-i+1|k-i+1} H(t_{k-i+1})^{\intercal}\right) \\
\hat{Q}_k &= \sum_{i=1}^{M} \lambda_i K_{k-i+1} \nu_{k-i+1} \nu_{k-i+1}^{\intercal} K_{k-i+1}
\end{aligned} \tag{2.11}$$

where μ_k is the residual at time t_k, which is the difference between actual measurement and its estimated value using the information available at time t_k, $\{\lambda_i, i = 1, 2, ..., M\}$ are forgetting factors, and $\sum_{i=1}^{M} \lambda_i = 1$. Note that using a moving window, as we place more weight on previous estimates, less fluctuation of $\hat{R}_k$ and $\hat{Q}_k$ will incur, and it takes longer for the model to capture changes in the system. Additionally, note that we replace $Q(t)$ in Step 1 with $\hat{Q}_k$ during the time interval $[t_{k-1}, t_k]$.

One of the traditional fault detectors used in conjunction with Kalman filter is the χ^2-detector (Brumback & Srinath, 1987; Bar-Shalom & Li, 1995; Geng & Wang, 2008). Since AEKF is a special type of Kalman filter, the χ^2-detector can be seamlessly applied to AEKF as well. Specifically, it constructs χ^2 test statistics to determine whether the new measurement falls into the gate region with the probability determined by the gate threshold σ, as shown in the following:

$$V_\gamma(k) = \{z : (z - \hat{z}_{k|k-1})^{\intercal} S_k^{-1} (z - \hat{z}_{k|k-1}) \leq \sigma\} \tag{2.12}$$

where $\hat{z}_{k|k-1}$ is the predicted value of measurement at time t_k. The χ^2 test statistics for the fault detector is defined as

$$\chi^2(t_k) = \nu_k^{\intercal} S_k^{-1} \nu_k. \tag{2.13}$$

For the χ^2 test to provide meaningful results, the innovation ν_k should be zero mean Gaussian distributed with covariance S_k. However, in reality, the innovation can follow a non-zero mean Gaussian distribution if there is bias in the background (e.g., due to non-zero mean process noise or imperfect model), as shown in Figure 2.1. This figure displays a scatter plot of normalized innovation generated from the training dataset in our experiments, when there exits a time delay and $\int_{t-\tau}^{t} a_n(r)dr$ in equation (2.6) is not zero. Moreover, in practice the variance of the normalized innovation is not normally distributed when the noise does not follow a Gaussian distribution. In the 2-dimensional case, the χ^2-detector defines a circular boundary with its center located at $(0, 0)$, i.e., the blue lines in Figure 2.1, which corresponds to the thresholding boundary of the χ^2-detector with

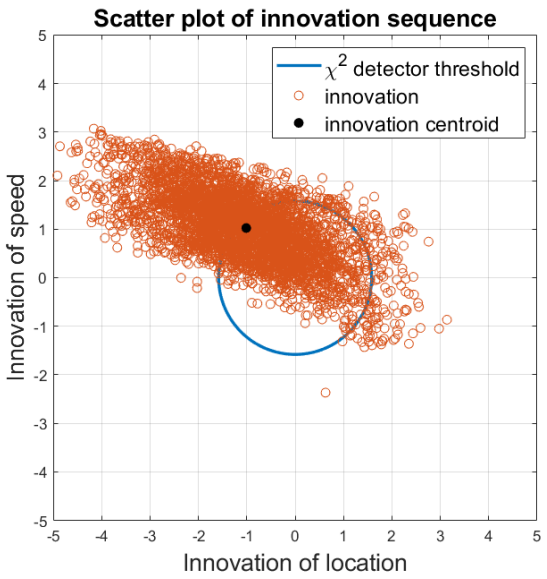


Figure 2.1: An example where the normalized innovation sequence is not zero mean due to time delay and imperfect model, where $\tau = 0.5$ second. The threshold σ of χ^2-detector is 2.5.

$\sigma = 2.5$. Therefore, in order to correctly detect anomalies, the χ^2-detector requires the data to be zero mean and normally distributed. In such a scenario, the χ^2-detector would not be a good detector since it will generate higher rates of false positives and false negatives. As such, the boundary should be shifted toward the true mean in order to achieve a "fair" boundary on both sides.

In the context of our problem, the approximation introduced in equation (2.6) can generate such a bias. This approximation assumes the value $\int_{t-\tau}^{t} a_n(r)dr$ to be zero. As such, unless the acceleration and deceleration rates during the period $[t\ ,\ t+\tau]$ sum to zero, or the time delay τ is equal to zero, the resulting bias would degrade the performance of the χ^2-detector. Consequently, we propose a novel approach to use one-class support vector machines (OCSVMs) (Schölkopf et al., 2001) to adaptively learn the normal boundary of the innovation sequence. Specifically, we train several OCSVM models using normal (i.e., non-anomalous) sensor data with different parameter values (i.e., anomaly percentages). We use the trained OCSVM models for detecting anomalies in real-time.

2.3.4 One Class Support Vector Machine

Let us define normalized innovation $\bar{\nu}$ at time t_k as:

$$\bar{\nu}(t_k) = S_k^{-\frac{1}{2}} \cdot \nu_k \tag{2.14}$$

Assume we have a training set $\mathcal{N}$, with l data points, $\{\bar{\nu}(t_1), ..., \bar{\nu}(t_L)\} \in \mathcal{L}$, sampled from a

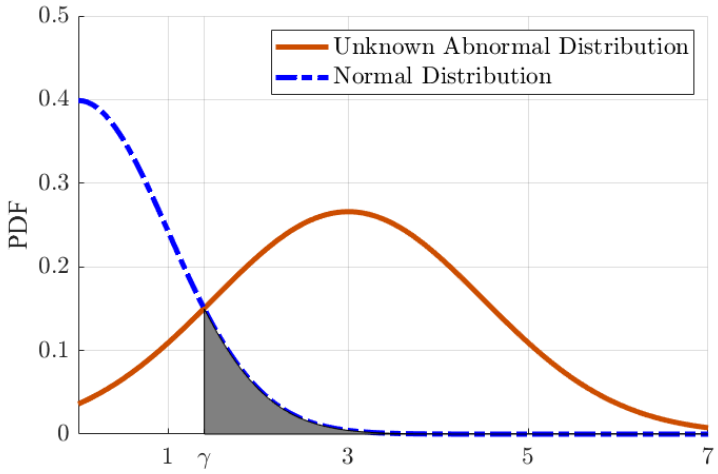


Figure 2.2: Distribution (PDF) of normal $|\bar{\nu}(t_k)|$ and the absolute value of unknown abnormal data. The shaded area indicates false positive beyond the threshold γ.

normal (i.e., non-anomalous) set. Let us define $\mathcal{K}$ as a kernel mapping function $\mathcal{L} \rightarrow \mathcal{F}$, where $\mathcal{F}$ represents the feature space. OCSVM solves the following quadratic program:

$$\begin{aligned} \min_{\omega \in \mathcal{F}, \boldsymbol{\xi} \in \mathbb{R}^l, \rho \in \mathbb{R}} \quad & \frac{1}{2}||\omega||^2 + \frac{1}{pl}\sum_j \xi_j - \rho \\ \text{subject to} \quad & \omega \cdot \Phi\left(\bar{\nu}(t_j)\right) \geq \rho - \xi_j \\ & \xi_j \leq 0 \end{aligned} \tag{2.15}$$

where p is a constant parameter in the range of $(0, 1)$, denoting the false positive rate of the decision boundary that classifies normal and anomalous sensor readings. Decision variables ω in model (2.15) define the most generalizable linear decision boundary in an infinite-dimensional space (created by the Gaussian kernel) to determine a region in the input space that encompasses at least $1 - p$ percentage of data points. Decision variables ξ_j are slack variables introduced to penalize the degree of violation of the constraint $(\omega \cdot \Phi\left(\bar{\nu}(t_j)\right)) \geq \rho$.

According to Proposition 3 in (Schölkopf et al., 2001), parameter p provides an upper bound on the fraction of outliers in the training dataset, and asymptotically equals the fraction of outliers out-of-sample, with probability 1, under certain conditions. We train M different OCSVM models, each model with a parameter p selected from the set $\{p_1, p_2, ..., p_M\}$ with $p_i < p_j, \forall i < j$. For a measurement sequence with dimension m, we compute the average of the normalized innovation

sequence over a window of N time intervals, up to time t_k,

$$\bar{\nu}_{avr}(t_k) = |\frac{1}{N}\sum_{i=k-N+1}^{k}\bar{\nu}(t_i)|_1 \tag{2.16}$$

where $|\cdot|_1$ is the L1-norm. For small values of $\bar{\nu}_{avr}(t_k)$, i.e., when the average of normalized innovation within the current time window is small, suggesting that AEKF performs well, we choose a trained OCSVM model with large value of p to ensure that we would detect even small variations and mark them as outliers. Following the same line of logic, when $\bar{\nu}_{avr}(t_k)$ is large, we choose an OCSVM model with small p to avoid unnecessary dismissal of data points where we are not certain enough that a point is truly an outlier. To that end, in order to determine the parameter p properly, we use the histogram of innovation values constructed from normal data to approximate the distribution of the innovation. Without loss of generality, we assume $\nu(t_k)$ is zero mean Gaussian distributed. For the case when it is not zero mean, we first subtract the mean value, and add it back after determining the value of p.

The tail of the probability density function (PDF) in Figure 2.2 represents drastic changes in data, and the center of the PDF represents smooth changes. The histogram of $|\bar{\nu}|$ is an approximation of the PDF of the normal training data. As shown in Figure 2.2, parameter γ controls the area of the shadow, and the PDF of the normal data is approximated by the histogram. We assume the absolute value of normal data as a random variable denoted as X with a certain distribution and domain $D(X)$. Then, given the number of training samples N_{train}, the number of outliers N_{ol} can be computed from

$$\begin{aligned} N_{ol} &= p \cdot N_{train} \\ &= \frac{|\{x : 1 - F(x) = p\}|_{mode}}{|\{x : x \in D(x)\}|_{mode}} \times N_{train} \\ &\approx \frac{|\{\bar{\nu} : |\bar{\nu}|_1 \geq \gamma\}|_{mode}}{N_{train}} \times N_{train} \end{aligned} \tag{2.17}$$

where $F(\cdot)$ is the CDF of X. As such, we have:

$$|\{\bar{\nu} : |\bar{\nu}|_1 \geq \gamma\}|_{mode} \approx p \cdot N_{train} \tag{2.18}$$

Finally, for M OCSVM models with parameters $\{p_1, p_2, ..., p_M\}$, when testing on a new data point, we have:

$$\begin{aligned} \bar{\nu}_{avr}(t_k) &\in [0, \gamma_1) \Rightarrow p_1 \\ \bar{\nu}_{avr}(t_k) &\in [\gamma_1, \gamma_2) \Rightarrow p_2 \\ &\cdots \\ \bar{\nu}_{avr}(t_k) &\in [\gamma_{M-1}, \infty) \Rightarrow p_M \end{aligned} \tag{2.19}$$

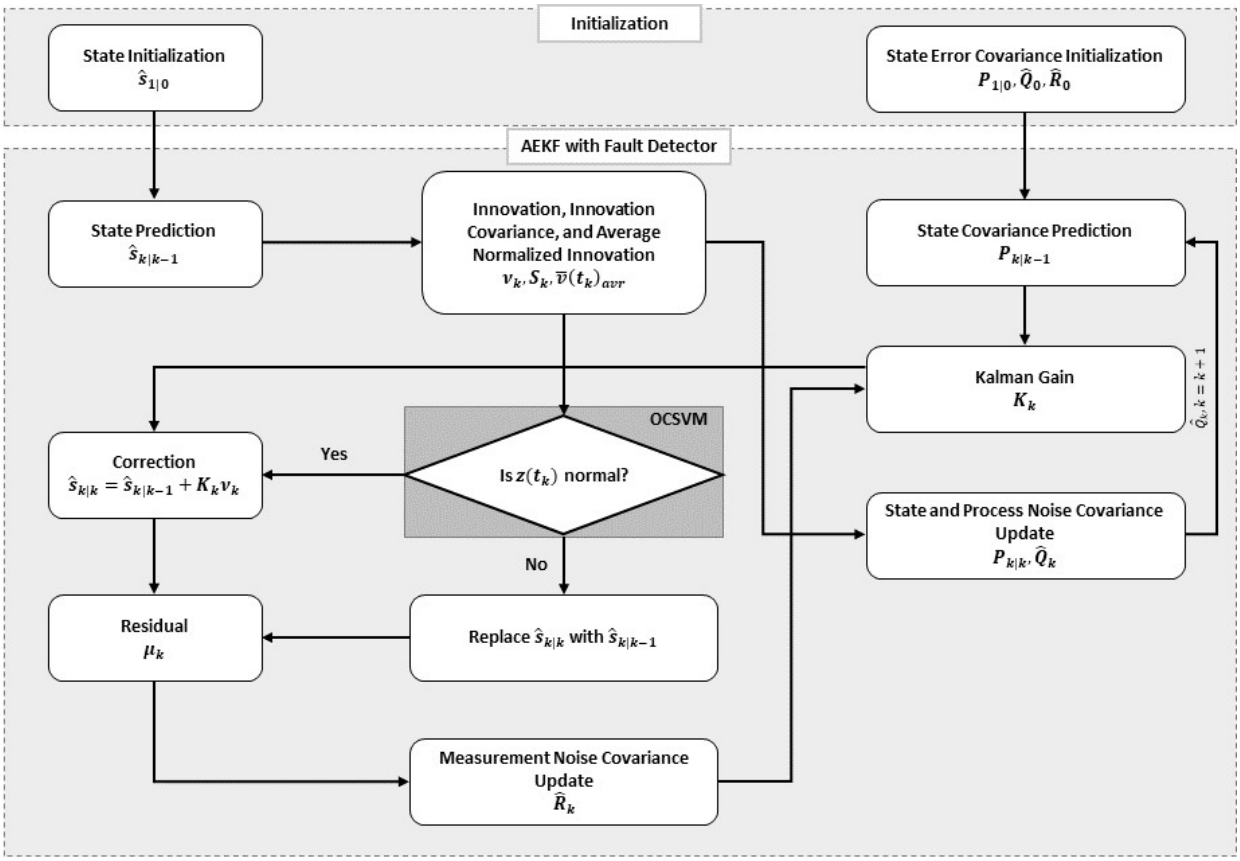


Figure 2.3: Implementation flowchart of the proposed algorithm.

In summary, assuming we have n measurements, Figure 2.3 summaries an implementation flowchart of the proposed algorithm, which combines AEKF and OCSVM to detect anomalies and recover the corrupt sensor readings. Specifically, at each time epoch, the following vehicle receives the measurements from both the leading vehicle and its own onboard sensors. The AEKF smooths the following vehicle's speed and location signal based on the motion model. Meanwhile the AEKF generates the innovation, which measures the discrepancy between the measurements and the prediction, and sends the innovation to the fault detector model for anomaly detection. The fault detector model consists of several OCSVM models and it can dynamically choose which one to use based on the average innovation. If there is no sensor anomaly detected, the innovation will be combined with the measurement at current time in order to generate an estimation. Otherwise, we do not trust the current sensor measurement and replace the estimation with the prediction, which will be used in the next time epoch.

2.3.5 Anomaly Model

The dataset for this chapter is generated by randomly adding anomalies to normal trajectory data, since there is no publicly available dataset on CAV trajectories that includes anomalies in sensor measurements. Specifically, we account for the four major anomaly types including short, noise, bias, and gradual drift. The detailed construction of the anomalies are as follows:

1. *Short:* The short anomaly type is simulated as a random Gaussian variable with mean and variance of 0 and c_1, respectively.
2. *Noise:* The noise anomaly type is simulated as a *sequence* of i.i.d. random Gaussian variables with length of l, mean of 0, and variance of c_2.
3. *Bias:* The bias anomaly type is simulated as adding a temporarily offset to the observation. We simulate the magnitude of the anomaly as Gaussian distributed with mean of 0 and variance of c_3. The duration of the sequence of bias anomaly is l.
4. *Gradual drift:* The gradual drift anomaly type is simulated by adding a linearly increasing/decreasing set of values to the base values of the sensors. Specifically, first we use a vector of linearly increasing values from 0 to m, where m is a uniformly distributed random variable in range of $[0, c_4]$. The duration of the sequence is l. We then use a Bernoulli random variable with probability 0.5 to generate one of the two outcomes of 1 and -1, by which we scale the sequence to generate increasing or decreasing drift, respectively.

Here $c_i, i \in \{1, 2, 3, 4\}$ is the parameter of distribution for each type of anomaly.

We inject all four types of anomalies into each sensor reading. We assume the *onset* of anomalous values in sensors occur independently. That is, we do not explicitly train OCSVM on a dataset containing interdependent sensor failures or systemic cyberattacks on vehicle sensors. However, this assumption does not preclude scenarios under which multiple sensors are under simultaneous attack.

Similar to the previous work (Van Wyk et al., 2019), we generate various datasets for our experiments with anomaly rate of α. In addition, we simulate anomalies to start at randomly selected times, lasting for random durations (if it applies to the anomaly type), affecting randomly selected sensors. These anomalies are then used to adjust the corresponding sensors' normal readings in the original dataset, which indicate the traveling location and speed of the CAV, making them anomalous. The pseudo code describing random generation of anomalies is presented in Algorithm 2.1. In the algorithm, $randi(L)$ denotes a discrete uniform distribution among integer numbers from 1 to L. Note that each anomaly type is equally likely to get selected when an anomaly is generated.

Algorithm 2.1: Anomaly Generation Process

$\alpha \leftarrow$ anomaly rate; $n \leftarrow$ number of sensors; $\boldsymbol{\zeta} = [\zeta_1, \zeta_2, ..., \zeta_n] \leftarrow \mathcal{U}(0, 1)$;
$L \leftarrow$ maximum duration of anomaly
for time epoch $t \in \mathcal{T}$ **do**
  for $i \in \{1, 2, \ldots, n\}$ **do**
    if no anomaly exists at t for the ith sensor **then**
      if $\zeta_i \leq \alpha$ **then**
        $l \leftarrow randi(L)$
        switch (Anomaly type selected according to probability distribution f_ω)
        case Short**:**
          Add 'short' type of anomaly with parameter c_1
        case Noise**:**
          Add 'noise' type of anomaly with duration l with parameter c_2
        case Bias**:**
          Add 'bias' type of anomaly with duration l with parameter c_3
        case Gradual Drift**:**
          Add 'gradual drift' type of anomaly with duration l with parameter c_4
        end switch
      end if
    end if
  end for
end for

2.4 Numerical Experiment

In this section we adopt a well-known car-following model, namely the Intelligent Driver Model (IDM), proposed by Treiber et al. (2000), to compare the anomaly detection performance of the traditional χ^2-detector and the OCSVM model. As mentioned in (Treiber & Kesting, 2012), since the IDM has no explicit reaction time and its driving behavior is given in terms of a continuously differentiable acceleration function, it describes more closely the characteristics of semi-automated driving by adaptive cruise control (ACC) than that of a human driver. However, it can easily be extended to capture the communication delay as described in the previous section. We also evaluate the impact of using the IDM motion model on anomaly detection performance. In order to evaluate system performance, we assume that the input vector containing the leading vehicle's information is not anomalous. When both the subject vehicle sensors and the leading vehicle's information present anomalous data, as long as the deviation of the state prediction and the measurement exceeds the threshold of the fault detector, the system still is able to detect sensor anomalies. However, when the deviation is not evident enough, e.g. under the rare circumstance when the attacker manipulates both the input vector and the measurement such that they remain consistent according to the car-following model, the fault detector will fail to detect those anomalies. However, such circumstance is not easy to occur since it requires knowledge from the part of the attacker on both the exact motion model as

well as how the fault detector threshold is dynamically updated.

Using the definition of state $s_n(t)$ and input $u_n(t)$ in the previous section, the IDM model with time delay τ can be described as the following:

$$\begin{aligned}\dot{x}_n(t) &= v_n(t) \\ \dot{v}_n(t) &= f_v(s_n(t-\tau), u_n(t-\tau)) \\ &= a\left(1 - \left(\frac{v_n(t-\tau)}{v_0}\right)^{\delta} - \left(\frac{s^*(v_n(t-\tau), v_n(t-\tau) - v_{n-1}(t-\tau))}{x_{n-1}(t-\tau) - x_n(t-\tau) - l_n}\right)^2\right)\end{aligned} \tag{2.20}$$

with

$$s^*(v_n, \Delta v_n) = s_0 + v_n T + \frac{v_n \Delta v_n}{2\sqrt{ab}}$$

where $a, b, \delta, v_0, s_0, T$ and l_n are model parameters. The state vector s_n and the input vector u_n both have dimension of 2. For detailed information on IDM refer to (Treiber et al., 2000). Following the typical parameter values of city traffic used in (Treiber & Kesting, 2012), we set the parameter values in our study as follows: $a = 1.0, b = 1.5, \delta = 4, v_0 = 33.75, s_0 = 2, T = 1.0$, $l_n = 5$, and define the measurement function $\mathcal{M}(\cdot)$ as:

$$\mathcal{M}(s) = H \cdot s = \begin{bmatrix} 1, 0 \\ 0, 1 \end{bmatrix} \cdot s \tag{2.21}$$

The data for this chapter is obtained from the research data exchange (RDE) database constructed as part of the Safety Pilot Model Deployment (SPMD) program (Bezzina & Sayer, 2014) funded by the US department of Transportation, and collected in Michigan. This program was conducted with the primary objective of demonstrating CAVs, with the emphasis on implementing and testing V2V and V2I communications technologies in real-world conditions. The program recorded detailed and high-frequency (10 Hz) data for more than 2,500 vehicles over a period of two years. The data features extracted from the SPMD dataset used in this study include the in-vehicle speed for one of the test vehicles with a trip length of 400 seconds (4000 samples) for training data, and 200 seconds (2000 samples) for testing data. As mentioned in Section 2.1, we assume that the vehicles are in ACC mode according to a car-following model, i.e. the IDM model. Therefore, as shown in Figure 2.4, we use the extracted speed data as the leading vehicle's speed v_{n-1}, and generate its location

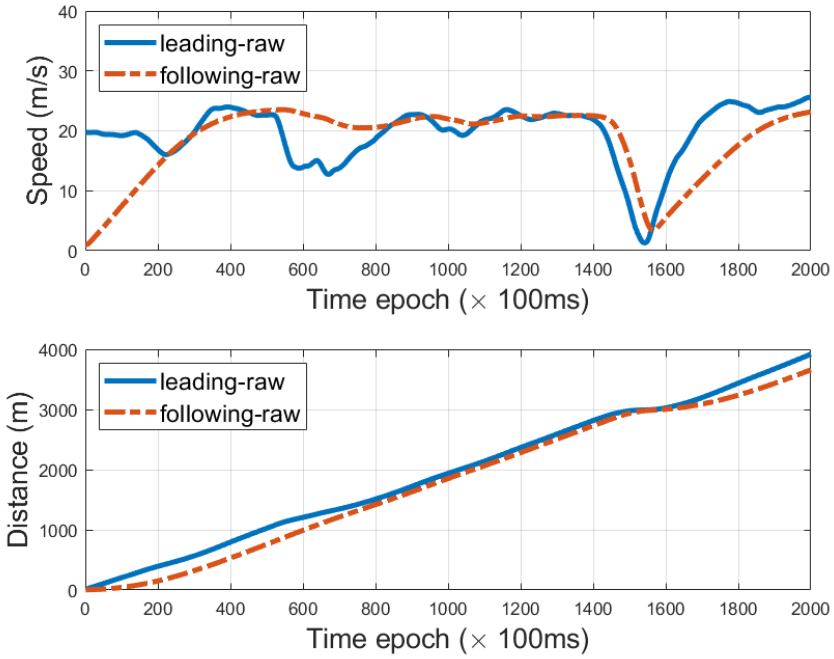


Figure 2.4: The synthetic car-following data using SPDP data under IDM model. The first and second plot are the speed and location of two vehicles respectively.

x_{n-1} and the following vehicle's state (x_n and v_n) as the baseline based on the following rules:

$$\begin{aligned} x_{n-1}(k+1) &= x_{n-1}(k) + v_{n-1}(k) \cdot \tau \\ x_n(k+1) &= x_n(k) + v_n(k) \cdot \tau \\ v_n(k+1) &= \Delta t \cdot f\left(x_n(k-\tau), v_n(k-\tau), x_{n-1}(k-\tau), v_{n-1}(k-\tau)\right) + v_n(k-\tau) + \epsilon \cdot \tau \end{aligned} \tag{2.22}$$

where ϵ is a random term that describes the uncertainty of the following vehicle's state. In our study we generate ϵ based on a uniformly distributed random variable within the range $[-0.1, 0.1]$. Furthermore, we add Gaussian white noise with variance 0.02 to the leading vehicle's baseline data. Since we want to test the detection performance, the noise variance should be smaller than the anomaly variance so that it would not be overpowered by the white noise. Note that adding white noise to the leading vehicle's baseline data is equivalent to using a Gaussian distributed random time delay factor of $\tilde{\tau}$ with mean τ.

To demonstrate the importance of incorporating the leading vehicle's information into the following vehicle's anomaly detection procedure, we implement our framework once using the IDM

car following model, and once without it, using the state-space model expressed in the following:

$$\begin{aligned} \dot{s}_n(t) &= \begin{bmatrix} e_2^\intercal \cdot s_n(t-\tau) \\ 0 \end{bmatrix} + \theta(t) \\ z_n(t_k) &= H s_n(t_k) + \eta(t_k),\ k \in \{0 \cup \mathbb{Z}^+\} \end{aligned} \tag{2.23}$$

where the process noise $\theta(t)$ accounts for the introduced error.

To measure the effectiveness of our two main contributions, i.e., incorporating a car following model into the AEKF motion model and using a OCSVM fault detector, we conduct sensitivity analysis over the motion model (i.e., with and without the IDM motion model), the anomaly detection methodology (i.e., the χ^2-detector and OCSVM), and the time delay (i.e., $\tau = \{0, 0.5, 1.5\}$). To evaluate the impact of changing models/parameters, we compute the Area Under the Curve (AUC) for each receiver operating characteristic (ROC) curve. The ROC curve is a graphical plot tool to illustrate the diagnostic ability of a binary classifier as its discrimination threshold is varied, and is created by plotting the true positive rate (sensitivity) against the false positive rate ($1-$ specificity) at various threshold settings. More specifically, we change the values of σ in our χ^2-detector, and the γ vector for the OCSVM. Note that the vector γ is a three-dimensional vector as we train and utilize four different OCSVM models.

The experiments are separately implemented into three scenarios, where scenario 1 contains a χ^2-detector without the IDM motion model, scenario 2 contains the χ^2-detector with the IDM model, and scenario 3 contains OCSVM with the IDM model. Each scenario is implemented under three experimental settings generated by varying the value of the anomaly parameter $c_i, i = \{1, 2, 3, 4\}$. More specifically, values of $c_i = 1$, $c_i = 0.1$, and $c_i = 0.05$ are used for settings 1, 2 and 3, respectively. This suggests anomalous readings become more subtle, and generally more difficult to detect, from setting 1 to setting 3. Lastly, the maximum duration of anomaly, L, is set to 20 for each setting.

Tables 2.1-2.3 present the AUC values of the three scenarios in our three experiment settings, with time delays of $\tau = 0$, 0.5, and 1.5 seconds, respectively. The experiments indicate that the IDM observer-based fault detection method provides significant improvement (up to 23%) compared with the performance of AEKF without the IDM model, regardless of the value of time delay. Additionally, we can see that OCSVM consistently achieves a better fault detection performance than the χ^2 detector. Results also indicate that there is a degeneracy of performance for each method as the parameter c_i becomes smaller. This observation is in line with intuition, since smaller c_i makes the anomaly more subtle and therefore harder to detect. Additionally, the trends of AUC values indicate that as we increase the time delay, the overall detection performance systemically deteriorates. This suggests that the time delay of the car-following model may have a negative

impact on the detection performance.

Table 2.1: AUC of three scenarios with $\tau = 0$ seconds.

	χ^2 without IDM	χ^2 with IDM	OCSVM with IDM
$c_i = 1$	0.9059	0.9723	0.9806
$c_i = 0.1$	0.7764	0.9453	0.9470
$c_i = 0.05$	0.7294	0.9228	0.9357

Table 2.2: AUC of three scenarios with $\tau = 0.5$ seconds.

	χ^2 without IDM	χ^2 with IDM	OCSVM with IDM
$c_i = 1$	0.9024	0.9703	0.9793
$c_i = 0.1$	0.7637	0.9402	0.9452
$c_i = 0.05$	0.7258	0.9118	0.9260

Table 2.3: AUC of three scenarios with $\tau = 1.5$ seconds.

	χ^2 without IDM	χ^2 with IDM	OCSVM with IDM
$c_i = 1$	0.8939	0.9701	0.9782
$c_i = 0.1$	0.7681	0.9201	0.9294
$c_i = 0.05$	0.7208	0.8875	0.8940

2.5 Conclusion

This chapter proposes an anomaly detection method to protect CAVs against anomalous sensor readings and/or malicious cyberattacks. We use an adaptive extended Kalman filter, informed by not only the vehicle's onboard sensors but also the leading vehicle's trajectory, in order to detect anomalous information. The well-known IDM car following model is used to incorporate the leading vehicle's information into AEKF. Lastly, to improve the anomaly detection performance, and given the fact that using AEKF the innovation is not normally-distributed, we replace the traditionally used χ^2-detector with an OCSVM model. We quantify the effect of these contributions in isolation, as well as in a combined model, by conducting experiments under three scenarios: (i) χ^2-detector without the IDM model, (ii) χ^2-detector with the IDM model, and (iii), OCSVM with the IDM model. Results show that the AEKF enhanced with OCSVM and the IDM model outperforms the traditional χ^2-detector-based anomaly detection used in conjunction with AEKF. Furthermore, our

results indicate that a model-based anomaly detection method that can incorporate the status of the lead vehicle can further improve the detection performance. More specifically, by utilizing the leading vehicle's information to inform the AEKF and using OCSVM for anomaly detection, the proposed method can not only effectively filter out the sensor noise in CAVs, but also detect the anomalous sensor values in real-time with better performance than that without utilizing leading vehicle's information. This high performance is showcased by high AUC values in our experiments. Moreover, we study the general relationship between the delay in receiving information and the performance of anomaly detection. We show that as the time delay of signal transmission (i.e., communication channel delay or sensor delay) becomes larger, the overall detection performance deteriorates.

The current study can be improved/expanded in multiple ways. First, the following vehicle's state and anomalous sensor values used in Section 2.4 are simulated, due to the paucity of ACC datasets with anomalies for CAVs. There exist multiple car-following datasets, but most of them were collected from human drivers. Although our study mainly focuses on the detection performance of our proposed method, it may be beneficial to directly collect ACC data from CAVs and calibrate the car-following model based on a real dataset, since the potential discrepancy between the car following model and the true traveling behaviour of the vehicle may introduce new challenges. Second, in Section 2.4, we assume the input vector containing leading vehicle's information is not anomalous. However, our proposed method can still detect anomalies without such an assumption, i.e., as long as the discrepancy between input vector and measurement are large enough. Note that we also assume that anomalous input vector are caused either by sensor failures or false injection attacks that can be described by four types of anomaly. Third, in this study for each vehicle we only utilize a single leading vehicle's information, whereas a connected vehicle can benefit from information shared by any number of connected vehicles within its communication range as well as the infrastructure. In Chapter 4, we study the impact of incorporating multiple sources of information (e.g., multiple vehicle) on the overall anomaly detection performance. Finally, a further direction can be the source identification of anomaly after detection.

CHAPTER 3

Anomaly Detection in Connected and Automated Vehicles Using an Augmented State Formulation

3.1 Introduction

A robust anomaly detection scheme designed for CAVs should satisfy three characteristics. First, any anomaly detection scheme should be able to effectively identify false negatives and false positives, both of which may result in severe negative outcomes as discussed in Chapter 2. To avoid false positives, the anomaly detection methods should be able to distinguish between anomalies and true unexpected changes in network conditions that may trigger unexpected responses from road users. Therefore, to reduce the number of false positives the anomaly detection scheme should be able to incorporate network-level information. To avoid false negatives, anomaly detection methods should be able to identify the noise, introduced by the vehicles' onboard sensor systems or the communication channel, to make sure decisions are not affected by it. Secondly, anomaly detection methods should not pose additional computational burden on a CAV's already constrained computational and energy resource. Finally, the anomaly detection techniques should be fast enough to be implementable in highly-dynamic traffic streams.

Anomalous sensor readings could be caused by different form of anomalies. In this chapter we adopt the same anomaly taxonomy from Chapter 2: (i) Short, which is a short-lived and sudden change in the observed data; (ii) Noise, which is a longer-term change (multiple successive readings) in variance of the observed data; (iii) Bias, which is an offset from the true sensor readings; (iv) Gradual drift, which is a gradual drift in the observed data and (v) Miss, which refers to missing data observations that could result from Denial of Service (DoS) attacks or sensor failures. In this chapter, we do not explicitly account for the anomaly type 'miss'; however, in practice, depending on its duration, the 'miss' anomaly type can be viewed as either 'instant' or 'bias' anomalies, where for a short or long period of time, respectively, the sensor readings are changed to zero.

The objective of this chapter is to develop an anomaly detection scheme that can incorporate network level information, does not require large computational resources, and can detect anomalies

in real-time. In Chapter 2, we developed a framework that combined an adaptive extended Kalman Filter, enhanced using a car-following motion model, and a data-driven fault detector to detect CAV anomalies in the presence of communication time delay. This chapter is an extension of Chapter 2, where we reformulate the problem using an augmented state in order to capture the bias caused by the time delay, and thereby improve the detection performance of the traditional χ^2-detector. Moreover, the new method proposed in this chapter can be seamlessly coupled with the CNN-KF framework proposed in (Van Wyk et al., 2019); experiments indicate improvements in performance by implementing the new method. We use the same assumptions as in Chapter 2 throughout this chapter: (i) vehicles follow their immediate upstream vehicle (referred to as the leader), according to a car-following model; and (ii) there is a time-delay associated with obtaining the leader's information (e.g., location and velocity).

The rest of the chapter is organized as follows: In Section 3.2 we provide a brief review of the existing literature in CAV anomaly detection. In Section 3.3 we introduce the problem formulation and our solution method. In Section 3.4 we conduct a case study based on a well-known car-following model and compare the performance with our previous method. Finally, in Section 3.5 we conclude the chapter.

3.2 Literature Review

Sensor failures and cyberattacks make two of the main contributors to anomalous sensor readings in CAVs. Sensor failure may result from environmental conditions (e.g. dense vegetation and tall buildings that block GPS satellite signals), sensor age, low battery supply, etc. (Ni et al., 2009). There has been a number of studies focusing on cyberattacks on CAVs (Petit & Shladover, 2015; Trippel et al., 2017; Faughnan et al., 2013). Experimental studies have demonstrated the vulnerability of a wide array of CAV sensors, e.g., speed, acceleration and location sensors, to cyberattacks or faults. For example, a false injection attack through the CAN bus or the on-board diagnostics (OBD) system of a CAV (Petit & Shladover, 2015) can result in any of the five anomaly types introduced in this chapter. As another example, Trippel et al. demonstrate how the in-vehicle acceleration sensor could be vulnerable to acoustic injection attacks (Trippel et al., 2017). Spoofing/jamming attacks on a CAV's GPS unit is another example of attackers inducing anomalies into sensor values (Faughnan et al., 2013).

Because of the potentially severe consequences of failing to detect anomalous sensor readings and/or anomalous data received through commutation channels, an increasing number of studies have focused on cyber security in CAVs. A wide range of frameworks founded on graph theory (Park et al., 2015), deep learning (Van Wyk et al., 2019), and game theory (Brahmi et al., 2019), among others, have been proposed for this purpose. In this chapter we propose a new anomaly

detection framework based on extended Kalman filter (EKF) with a car-following motion model and an augmented state space. This framework offers an anomaly detection method that (i) directly targets reducing the number of false positive errors in anomaly detection; (ii) does not require significant computational resources; and (iii) can be executed in real-time, making it suitable for dynamic traffic environments. This framework is an extension of Chapter 2, where we focused on detection of anomalous sensor readings and recovery of the corrupt signals. Similar to Chapter 2, we use a car-following model as the motion model of an EKF in order to use the trajectory of the subject vehicle's leading vehicle, thereby capturing some network-level information. The leading vehicle's information is received through basic safety messages (BSM) enabled by V2V communications. When using the leading vehicle's information to inform the motion of the subject vehicle, the time-delay in the communication channel could create theoretical challenges that would exclude the use of traditional fault-detectors, such as the χ^2 detector. In Chapter 2, we addressed this issue by introducing a data-driven fault detector. In this chapter we adopt an augmented state for EKF, i.e. augmented state extended Kalman filter (ASEKF), which makes it possible to use a traditional χ^2 detector. Additionally, in this chapter we analyze the impact of stochastic time delay in receiving the leading vehicle's information. Our experiments show a boost of detection performance of χ^2-detector with augmented state formulation. Additionally, they show a lower mean squared error (MSE) compared with our previous formulation under time delay.

3.3 Solution Methodology

In this section, we first discuss how to reformulate a car-following model into a continuous motion model with time delay using an augmented state. Augmenting the state allows us to compensate for the potential bias caused by the approximation process and model inaccuracy. Based on that, we then formulate a new continuous nonlinear state-space model with discrete measurements based on a car-following model, where the continuous state-transition model represents the intrinsic nature of a vehicle's response to the actions of its immediate leader, and the discrete measurement model represents the discrete nature of sensor sampling. Next, we analyze the impact of stochastic time delay on the system. Finally, we apply the augmented state extended Kalman filter (ASEKF) to the state-space model, and use the resulting ASEKF model in conjunction with a χ^2-detector to find anomalies.

3.3.1 Car-Following Model with Time Delay

We use a typical car-following model as described in (Treiber & Kesting, 2012):

$$\begin{aligned} d_n(t) &= x_{n-1}(t) - x_n(t) \\ \dot{d}_n(t) &= v_{n-1}(t) - v_n(t) \\ \dot{v}_n(t) &= f(v_n(t-\tau), d_n(t-\tau), \dot{d}_n(t-\tau)) \end{aligned} \tag{3.1}$$

where $\dot{v}_n(t)$, $v_n(t)$, $x_n(t)$ are the acceleration, speed, and location of the nth vehicle, to which we refer as the subject vehicle or the following vehicle, respectively. In equation (3.1), $d_n(t)$ and $\dot{d}_n(t)$ are the headway and the speed difference between the subject vehicle and its leading vehicle (also referred to as the leader), i.e. the $(n-1)$th vehicle, respectively. Parameter τ denotes time delay, intended to capture the time lapsed between the moment the leader performs an action, to the moment the subject vehicle acts in response. For now we consider a constant time delay τ; however, later we relax the constant assumption by considering a stochastic time delay. Function f is the stimulus function.

Following the steps in Section 2.3, we define the state vector of the nth vehicle in continuous time as $s_n(t) = [x_n(t), v_n(t)]^\intercal \in \mathbb{R}^2$, and the input vector containing information received from the leading vehicle as $u_n(t) = [x_{n-1}(t), v_{n-1}(t)]^\intercal$. Consequently, we can recast equation (3.1) as a function of $s_n(t)$, producing a car-following model that maps the state into an actionable decision for the subject vehicle:

$$\dot{v}_n(t) := f_v\left(s_n(t-\tau), u_n(t-\tau)\right) \tag{3.2}$$

where f_v denotes the stimulus function describing velocity in a continuous sate space.

3.3.2 State-Space Model with Continuous State and Discrete Measurement

In this section we define a state-space model with a continuous state-transition model and discrete measurements, which will be used for ASEKF. Based on previous definition of the state vector $s_n(t)$, the state-transition model satisfies the following differential equation:

$$\dot{s}_n(t) = \begin{bmatrix} \dot{x}_n(t) \\ \dot{v}_n(t) \end{bmatrix} = \begin{bmatrix} e_2^\intercal s_n(t) \\ f_v(s_n(t-\tau), u_n(t-\tau)) \end{bmatrix} \tag{3.3}$$

where $e_2 = [0, 1]^\intercal$ is the standard basis vector.

Ideally when $\tau = 0$, the state-space model in equation (3.3) satisfies the Markovian property, allowing for applying ASEKF. However, in practice, τ is usually a nonzero value affected by various factors, e.g., communication network or data processing delay, etc.. As such, in practice ASEKF

cannot be applied to equation (3.3), since the derivative of the state vector is determined by multiple previous state vectors.

In order to apply ASEKF, we approximate equation (3.3) in the following way: Based on the bounded acceleration assumption, we can obtain a delay differential equation (DDE), describing the delayed state-transition model:

$$\begin{aligned}
\dot{s}_n(t) &= \begin{bmatrix} \dot{x}_n(t) \\ \dot{v}_n(t) \end{bmatrix} \\
&= \begin{bmatrix} e_2^\intercal s_n(t-\tau) + \int_{t-\tau}^{t} a_n(r)dr \\ f_v(s_n(t-\tau), u_n(t-\tau)) \end{bmatrix} \\
&= \begin{bmatrix} 1,0,1 \\ 0,1,0 \end{bmatrix} \times \begin{bmatrix} \tilde{e}_2^\intercal s_n(t-\tau) \\ f_v(s_n(t-\tau), u_n(t-\tau)) \\ \int_{t-\tau}^{t} a_n(r)dr \end{bmatrix} \\
&= \begin{bmatrix} 1,0,1 \\ 0,1,0 \end{bmatrix} \times \begin{bmatrix} \tilde{e}_2^\intercal \tilde{s}_n(t-\tau) \\ f_v(\tilde{e}_{12}^\intercal \tilde{s}_n(t-\tau), u_n(t-\tau)) \\ \tilde{e}_3^\intercal \tilde{s}_n(t-\tau) \end{bmatrix}
\end{aligned} \tag{3.4}$$

where $a_n(t)$ is the acceleration of the nth vehicle at time t, and $\tilde{s}_n(t) = [x_n(t), v_n(t), \delta_n(t)]^\intercal$ denotes the augmented state vector of $s_n(t)$ with augmented state $\delta_n(t) = \int_{t-\tau}^{t} a_n(r)dr$, $\tilde{e}_{12} = [\tilde{e}_1, \tilde{e}_2, 0]^\intercal$, and $\tilde{e}_i \in \mathbb{R}^3$ is the standard basis vector with i-th element equal to 1 and 0 otherwise. Since $\delta_n(t)$ is unknown, we assume it is a constant or a random variable with small variance. Thus we have $\dot{\delta}_n(t) \approx 0$.

The state-transition model with respect to the augmented state vector $\tilde{s}_n(t)$ can be presented as follows:

$$\begin{aligned}
\dot{\tilde{s}}_n(t) &= \begin{bmatrix} \dot{x}_n(t) \\ \dot{v}_n(t) \\ \dot{\delta}_n(t) \end{bmatrix} \\
&= \begin{bmatrix} \tilde{e}_2^\intercal \tilde{s}_n(t-\tau) + \tilde{e}_3^\intercal \tilde{s}_n(t-\tau) \\ f_v(\tilde{e}_{12}^\intercal \tilde{s}_n(t-\tau), u_n(t-\tau)) \\ 0 \end{bmatrix} + \theta(t) \\
&= \mathcal{T}(\tilde{s}_n(t-\tau), u_n(t-\tau)) + \theta(t)
\end{aligned} \tag{3.5}$$

where $\mathcal{T}(\cdot)$ is the state-transition model and $\theta(t)$ is the process noise, which accounts for the error introduced by the approximation and model inaccuracy.

Finally, using the new augmented state vector $\tilde{s}_n(t)$, we obtain a continuous-time state-transition

model with discrete-time measurement as the following:

$$\begin{aligned} \dot{\tilde{s}}_n(t) &= \mathcal{T}(\tilde{s}_n(t-\tau), u_n(t-\tau)) + \theta(t) \\ z_n(t_k) &= \mathcal{M}(\tilde{s}_n(t_k)) + \eta(t_k),\ k \in \{0 \cup \mathbb{Z}^+\} \end{aligned} \tag{3.6}$$

where $\mathcal{M}(\cdot)$ is the measurement function, $z_n(\cdot)$ denotes sensor reading of the leading vehicle, $\eta(t_k)$ is the observation noise, which is assumed to be mutually independent with the process noise, $t_{k+1} = t_k + \Delta t,\ k \in \{0 \cup \mathbb{Z}^+\}$, and Δt is the sampling time interval for sensors.

3.3.3 Stochastic Time Delay

Now we consider a more general case where the time delay τ_s is not a known constant. We assume that τ_s is a linear model:

$$\tau_s = \tau + \kappa \tag{3.7}$$

where κ is zero mean truncated Gaussian distributed with variance σ_1^2, and within the intervals (b_l, b_u). Therefore we have $E(\tau_s) = \tau$. We also assume the leading vehicle's trajectory obeys a linear model, i.e.,

$$\begin{cases} \dot{x}_{n-1}(t) = v_{n-1}(t) \\ \dot{v}_{n-1}(t) = a_{n-1}(t). \end{cases} \tag{3.8}$$

Since the leading vehicle's motion model is unknown, we assume that the acceleration of the leading vehicle, a_{n-1}, is a Gaussian process with zero mean and variance σ_2^2.

Proposition 3.1. *A random time delay τ_s is equivalent to adding noise into the input vector $u_n(t-\tau)$ with fixed time delay τ.*

Proof. By integrating $\dot{x}_{n-1}(t-\tau_s)$, we have

$$\begin{aligned} x_{n-1}(t-\tau_s) &= \int_0^{t-\tau_s} v_{n-1}(\xi)d\xi \\ &= \int_0^{t-\tau-\kappa} v_{n-1}(\xi)d\xi \\ &= \int_0^{t-\tau} v_{n-1}(\xi)d\xi - \int_{t-\tau-\kappa}^{t-\tau} v_{n-1}(\xi)d\xi \\ &= x_{n-1}(t-\tau) + \epsilon_1(t-\tau;\kappa) \end{aligned} \tag{3.9}$$

where $\epsilon_1(t-\tau;\kappa) = -\int_{t-\tau-\kappa}^{t-\tau} v_{n-1}(\xi)d\xi$. Therefore using random time delay is equivalent to adding noises $\epsilon_1(t-\tau;\kappa)$ into state $x_{n-1}(t-\tau)$. Note that $\epsilon_1(t-\tau;\kappa)$ does not necessarily have

zero mean:

$$\mathbb{E}_\kappa[\epsilon_1(t-\tau;\kappa)] = \int_{b_l}^{b_u} \tilde{\phi}(\iota;0,\sigma_1,b_l,b_u) \int_0^{-\iota} v_{n-1}(\xi+t-\tau)d\xi d\iota \tag{3.10}$$

where where $\tilde{\phi}(\iota;\mu,\sigma_1,b_l,b_u)$ represents the probability density function of truncated normal distribution,

$$\tilde{\phi}(\iota;\mu,\sigma_1,b_l,b_u) = \frac{1}{\sigma_1}\frac{\phi(\frac{\iota-\mu}{\sigma_1})}{\Phi(\frac{b_u-\mu}{\sigma_1}) - \Phi(\frac{b_l-\mu}{\sigma_1})} \tag{3.11}$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ are the probability density function and cumulative density function of the standard normal distribution, respectively.

Similarly, for v_{n-1}, using random time delay is equivalent to adding noise $\epsilon_2(t-\tau;\kappa)$ into $v_{n-1}(t-\tau)$:

$$v_{n-1}(t-\tau_s) = v_{n-1}(t-\tau) + \epsilon_2(t-\tau;\kappa) \tag{3.12}$$

where $\epsilon_2(t-\tau;\kappa)$ does not necessarily have zero mean.

Therefore, the input vector can be expressed as a linear model:

$$u_n(t-\tau_s) = u_n(t-\tau) + \epsilon(t-\tau;\kappa) \tag{3.13}$$

where $\epsilon(t-\tau;\kappa) = [\epsilon_1(t-\tau;\kappa), \epsilon_2(t-\tau;\kappa)]^\intercal$. □

When the time delay of the input vector is stochastic, the continuous-time state-transition model with discrete-time measurement is as follows:

$$\begin{aligned} \dot{\tilde{s}}_n(t) &= \mathcal{T}(\tilde{s}_n(t-\tau), u_n(t-\tau_s)) + \theta(t) \\ z_n(t_k) &= \mathcal{M}(\tilde{s}_n(t_k)) + \eta(t_k),\ k \in \{0 \cup \mathbb{Z}^+\} \end{aligned} \tag{3.14}$$

Because of the term $\epsilon(t-\tau;\kappa)$, plugging equation (3.13) into (3.14) could cause a non-zero mean for $\theta(t)$, depending on the specific formulation of the car-following model. As mentioned in the next section, the existence of bias in the process noise $\theta(t)$ could deteriorate the performance of the traditional χ^2 fault detector, whereas using ASEKF can mitigate this issue.

3.3.4 Augmented State Extended Kalman Filter with Fault Detector

In order to smooth the CAV sensor noise, we apply ASEKF to the state-space model (3.6). The ASEKF contains the same predict and update routine as EKF in (2.8)-(2.10).

One of the classic fault detectors used in conjunction with Kalman filter is the χ^2-detector (Bar-Shalom & Li, 1995). Since ASEKF is essentially a special type of Kalman filter, the χ^2-detector can be seamlessly applied to ASEKF as well. Specifically, it constructs χ^2 test statistics to determine whether the new measurement falls into the gate region with the probability determined by the gate threshold γ in equation (3.15),

$$V_\gamma(k) = \{z : (z - \hat{z}_{k|k-1})^\intercal S_k^{-1}(z - \hat{z}_{k|k-1}) \leq \gamma\} \tag{3.15}$$

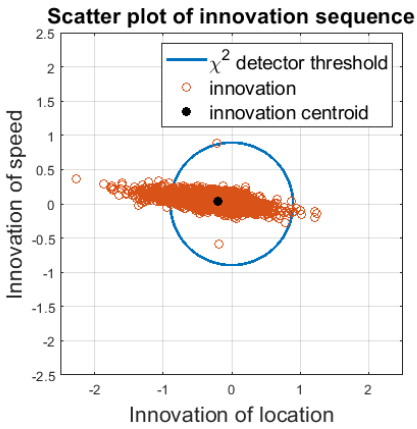
where $\hat{z}_{k|k-1}$ is the predicted value of measurement at time t_k. The χ^2 test statistics for the fault detector is defined as $\chi^2(t_k) = \nu_k^\intercal S_k^{-1} \nu_k$. In order to make the χ^2 test provide meaningful results, the innovation ν_k should be zero mean Gaussian distributed with covariance S_k; therefore, we combine it with ASEKF, which could compensate the potential bias caused by time delay.

In summary, we combine ASEKF and χ^2 fault detector to detect anomalies and recover the corrupt sensor readings. Specifically, at each time epoch, the subject vehicle receives the measurements from both the leading vehicle and its own onboard sensors. ASEKF uses the car-following motion model to smooth the following vehicle's speed and location signals. During the smoothing process, the ASEKF generates the innovation, which is the the discrepancy between the measurements and the prediction, and sends the innovation to the fault detector model for anomaly detection. If there is no sensor anomaly detected, the innovation will be combined with the measurement at the current time epoch in order to generate an estimation. Otherwise, the prediction would replace the estimation, which will be used in the next time epoch.

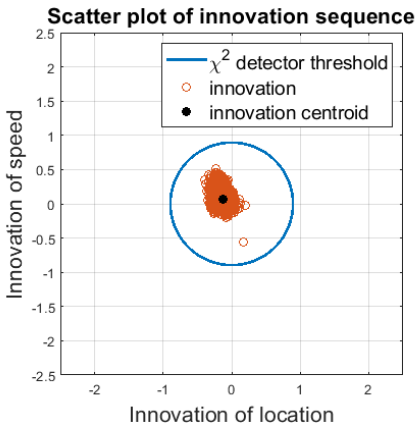
3.4 Numerical Experiment

In this section, we again use the Intelligent Driver Model (IDM) proposed by Treiber et al. (2000) as our car-following model of choice, and implement our framework to compare the anomaly detection performance of the χ^2-detector in conjunction with EKF and ASEKF. Since the IDM has no explicit reaction time and its driving behavior is given in terms of a continuously differentiable acceleration function (Treiber & Kesting, 2012), it is suitable for modeling semi-automated (compared to human) driving. However, it can be easily extended to capture the communication delay, as described in the previous section. Note that in order to evaluate system performance, we assume that the input vector containing the leading vehicle's information is not anomalous. We also use a truncated Gaussian distribution with mean τ and variance 1 to generate stochastic time delay.

Using the same model in (2.20) Following the typical parameter values of city traffic used in (Treiber & Kesting, 2012), we set the parameter values in our study as follows: $a = 1.0, b = 1.5, \delta = 4, v_0 = 33.75, s_0 = 2, T = 1.0$, $l_n = 5$, and define the measurement function $\mathcal{M}(\cdot)$ of



(a) Scatter plot of EKF.



(b) Scatter plot of ASEKF.

Figure 3.1: Example of a normalized innovation sequence being non-zero mean, where $\tau = 1.5$ seconds. The threshold σ of χ^2-detector is 0.8.

ASEKF in equation (3.14) as:

$$\mathcal{M}(\tilde{s}) = H \cdot \tilde{s} = \begin{bmatrix} 1, 0, 1 \\ 0, 1, 0 \end{bmatrix} \cdot \tilde{s} \tag{3.16}$$

We use the same dataset as in Chapter 2, which is based on Safety Pilot Model Deployment (SPMD) program (Bezzina & Sayer, 2014) funded by the US department of Transportation, and collected in Michigan. The anomalies are randomly generated with 5% anomaly rate and injected into the raw dataset using Algorithm 2.1 in Chapter 2.

We implement two sets of models, where model 1 is composed of the traditional EKF with an IDM motion model and a χ^2-detector, and model 2 is composed of the ASEKF with the IDM model and the χ^2-detector. Each model is implemented under three experimental settings generated by varying the value of the anomaly parameter $c_i, i = \{1, 2, 3, 4\}$ of Algorithm 2.1, which depending on the anomaly type describes the variance or magnitude of different types of anomalies. More specifically, values of $c_i = 1$, $c_i = 0.5$, and $c_i = 0.1$ are used for settings 1, 2 and 3, respectively. This suggests anomalous readings become more subtle, and generally more difficult to detect, from setting 1 to setting 3. We also conduct the each experimental setting under 3 different time delays of $\tau = \{0, 0.5, 1.5\}$ (seconds). Lastly, the maximum duration of an anomaly, L, is set to 20 for each setting. In order to evaluate the impact of changing models/parameters, we compute the Area Under the Curve (AUC) for each receiver operating characteristic (ROC) curve.

Table 3.1 presents the AUC values, averaged over 20 random instances, of all nine scenarios that are generated by changing the intensity of anomalies ($c_i = \{1, 0.5, 0.1\}$) and time delay ($\tau = \{0, 0.5, 1, 5\}$). For each scenario, we use a paired t-tests with a 5% significant level to

Table 3.1: AUC of detection performance and its standard deviation across 20 different executions for two models, at the anomaly rate of 5% and in the presence of all anomaly types. P-values indicate statistical significance at 5% level using paired t-test between the detection performance of each pair of models except between for the scenarios with $c_i = 1$ and $\tau = 0.5$.

	Model 1			**Model 2**		
τ	**0.0**	**0.5**	**1.5**	**0.0**	**0.5**	**1.5**
$c_i = 1$	0.984 $\pm$0.015	0.931 $\pm$0.028	0.855 $\pm$0.034	0.983 $\pm$0.022	0.945 $\pm$0.030	0.902 $\pm$0.038
$c_i = 0.5$	0.972 $\pm$0.020	0.917 $\pm$0.033	0.799 $\pm$0.391	0.969 $\pm$0.024	0.928 $\pm$0.034	0.844 $\pm$0.046
$c_i = 0.1$	0.871 $\pm$0.029	0.718 $\pm$0.031	0.576 $\pm$0.034	0.867 $\pm$0.035	0.759 $\pm$0.039	0.731 $\pm$0.058

determine whether there is a statically significant difference in the AUC values of Models 1 and 2. Results indicate that when there is no time delay, EKF with χ^2-detector (model 1) consistently achieves a better anomaly detection performance than the ASEKF with χ^2-detector (model 2). This is because when time delay is zero, there is no need to compensate for potential bias. Under such circumstances, the augmented state variable behaves similarly to additional noise added to the first element of the state variable (i.e., geo-location) as shown in equation (3.5), causing a lower detection performance. Results indicate that model 2 outperforms model 1 in all scenarios where $\tau > 0$, by capturing the the bias that is generated due to time-delay. It should be noted that in practice, a zero time delay may rarely occur, indicating that the augmented state formulation would perform better in practice. Results also indicate that there is a degeneracy of performance for each method as the parameter c_i becomes smaller. This observation is in line with intuition, since smaller c_i makes the anomaly more subtle and therefore harder to detect. Additionally, the trends of AUC values indicate that as we increase the time delay, the overall detection performance systemically deteriorates. This suggests that the time delay in general may have a negative impact on detection performance.

In order to investigate the effect of using the augmented state formulation, we find the mean innovation value for the two state variables (i.e., location and speed) for each model under the three time delay settings with zero percentage of anomaly, as presented in Table 3.2. As shown in the TABLE 3.2, ASEKF can significantly decrease the background bias in the first state variable, i.e. location of the subject vehicle, when there is a time delay. Similar to previous results, when the time delay is zero, the performance of ASEKF deteriorates, since no potential bias need to be compensated for, and the augmented state variable only introduces more randomness in the system. As time delay becomes larger, the effect of bias correction becomes more prominent. Table 3.2 also shows the mean squared error (MSE) in the innovation of the two models, where MSE is calculated

Table 3.2: Mean innovation for each state variable and the MSE for the two models at zero anomaly rate. MSE is calculated as the squared root of sum of the MSEs for the two state variables.

τ	**0.0**	**0.5**	**1.5**
Model 1			
Mean innovation value of location	-0.003	-0.146	-0.436
Mean innovation value of speed	0.022	0.028	0.062
MSE	7.12E-04	0.033	0.098
Model 2			
Mean innovation value of location	-0.002	-0.033	-0.097
Mean innovation value of Speed	0.024	0.036	0.095
MSE	8.43E-04	0.032	0.095

as the squared root of sum of the MSEs for the two state variables. further confirming our previous observations–as we increase the time delay, the reduction in MSE under model 2 (ASEKF) becomes larger compared with that under model 1 (EKF). Figures 3.1(a) and 3.1(b) display the scatter plots of normalized innovation for both EKF and ASEKF, generated from the training dataset in our experiments, when there exits a time delay and $\int_{t-\tau}^{t} a_n(r)dr$ in equation (3.4) is not zero mean. In the 2-dimensional case, the χ^2-detector defines a circular boundary with its center located at $(0, 0)$, i.e., the blue lines in Figure 3.1, which corresponds to the thresholding boundary of the χ^2-detector with $\sigma = 0.8$. We can see a significant reduction in the mean and variance of the normalized innovation of the first state variable in ASEKF. This figure also shows that the innovation sequence of ASEKF resembles a normal distribution more closely, indicating the suitability of the χ^2-detector.

3.5 Conclusion

The goal of this chapter is to develop a framework to detect anomalous sensor readings and/or data received through V2V or V2I communications in CAVs. The proposed framework introduces a computationally light-weight anomaly detection tool that attempts to minimize false positives and negatives by incorporating context from the driving environment, obtained through V2V communications. The delay in the communication channel can introduce theoretical challenges that preclude using existing anomaly detection tools. As such, we introduce an augmented state extended Kalman filer (ASEKF) that is informed by not only the vehicle's onboard sensors but also the leading vehicle's trajectory. In our case study, we use the well-known intelligent driver car following model to incorporate the leading vehicle's information. In conjunction with ASEKF, we use the classic χ^2-detector to detect five types of anomalies, which encapsulate general sensor faults and/or cyberattacks in CAVs. We also analyze the effect of stochastic time delay on the detection performance. We quantify the effect of these contributions by conducting experiments

under two scenarios: χ^2-detector with an IDM motion model and EKF, and χ^2-detector with an IDM motion model and ASEKF. Results indicate that in the presence of time delay, the second model outperforms the first. Furthermore, results show that in the presence of time delay, ASEKF can decrease the overall innovation MSE by compensating for the background bias. Potential extension includes exploration of different types of attack/anomaly and learning-based detector/classifier.

CHAPTER 4

Anomaly Detection and String Stability Analysis in Connected Automated Vehicular Platoons

4.1 Introduction

In order to ensure that a CAV can safely and effectively navigate the network, it needs access to robust and accurate data streams. As a result, any anomalous sensor data, if undetected, can greatly imperil the decision making process of CAVs. Either a malicious cyberattack or a sensor fault can result in an anomaly in CAV sensors. Moreover, with the presence of measurement noise, it is crucial to detect any anomalous sensor readings in real-time and accurately estimate the true state of the CAV meanwhile to ensure the safety of the CAV driving.

Anomalous sensor readings can be caused by a variety of reasons and manifest in different ways. In this chapter, we adopt the taxonomy of sensor failure/attack in Chapters 2 - 3 which is also provided by Van Wyk et al. (2019) and Watts et al. (2022) including 'bias', 'gradual drift', 'noise', 'short', and 'miss'. Specifically, 'bias' and 'gradual drift' impose a constant offset and a gradual drift from the actual sensor readings, respectively. The anomaly type 'noise' represents a duration of change of variance in the observed readings. The anomaly type 'short' refers to a abrupt and short-lived change, and 'miss' is a short- or long-term missed observation in the sensor readings. The 'miss' anomaly type can be viewed as a special case of either 'short' or 'bias' anomaly types, depending on its duration, with the sensor reading being zero.

Meanwhile, connectivity enables a CAV to receive information from its surrounding vehicles and infrastructures. However, wireless communications and some forms of cyberattacks (e.g., the jamming attack) can introduce time delays to the receiver, which can significantly impact the state estimation accuracy and traffic stability, causing delay and chaos, and even leading to fatal crashes. Therefore, it is imperative to take time delay into consideration during state estimation and anomaly detection processes, and investigate the potential impacts of cyberattacks on traffic stability.

Figure 4.1 shows an illustrative scenario of platoon with 3-predecessor following information flow topology. Other topologies discussed in (Feng et al., 2019) such as predecessor following,

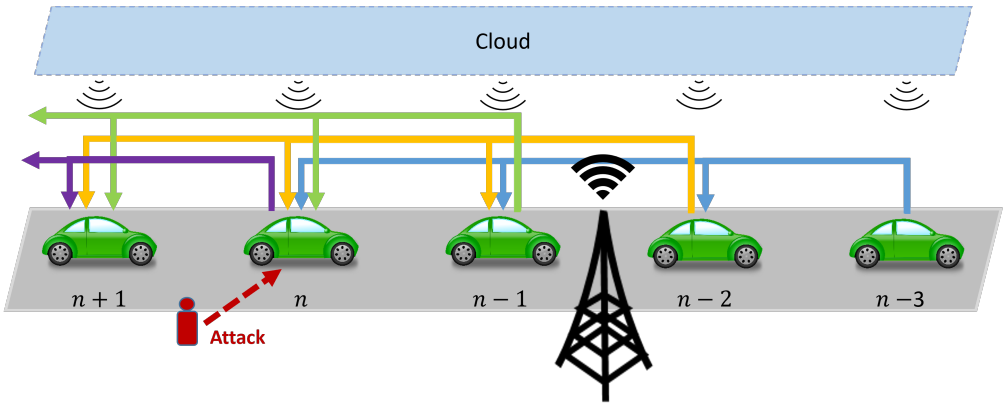


Figure 4.1: An example of platoon with 3-predecessor following information flow topology and with cloud information.

predecessor leader following, etc. can also be addressed in our framework. A malicious attacker can conduct cyberattacks to one or multiple vehicles in the platoon by manipulating the sensor reading of the compromised vehicle. We assume each vehicle is equipped with an anomaly detector and is capable of recovering the true sensor reading from the cloud once it successfully detects an abnormal sensor reading. The information from the cloud used for the recovery of anomalous sensor readings can be obtained from a road side unit (RSU), when possible, which monitors the vehicle trajectory and is assumed to be free from attacks.

In this chapter, we develop a holistic detection framework to improve the safety and security of CAV systems. Our framework combines sensor signal filtering and observer-based anomaly detection methods. Specifically, we consider anomaly detection in a platoon, where the ego vehicle can receive information from multiple sources, including other platoon members. We develop a general platooning framework for modeling a CAV's longitudinal dynamics under different types of cyberattacks. We use an augmented state extended Kalman filter (ASEKF) to estimate vehicle states from observed sensor readings of a CAV based on a nonlinear platooning model. In using the platooning model, the ego vehicle (i.e., the following CAV) leverages information from its leading vehicles to detect sensor anomalies by utilizing a set of offline-trained One Class Support Vector Machine (OCSVM) models. Stochastic and heterogeneous communication time delay factors are considered in the platoon dynamics to make it more congruent with real-world applications. Furthermore, we propose a novel definition of string stability, namely pseudo string stability, which represents the degree of string stability under model uncertainty. We establish the relationship between cyberattack detection rate and pseudo string stability, and identify the critical detection rate for maintaining pseudo string stability.

The contributions of this chapter can be summarized as threefold:

1. We extend the longitudinal platoon dynamics from Wang et al. (2020a), by considering a

more realistic setting that accounts for heterogeneous time delay instead of a homogeneous time delay in previous literature. With heterogeneous time delay, we assume that the onboard sensor readings, i.e., the state vector, and the communication channel, i.e., the input vector, may experience different time delays. More specifically, we propose a modelling framework that supports a comprehensive analysis of CAV platoon performance under cyberattacks, including the false injection attack, the jamming attack, etc. We further consider the stochastic time delay setting and investigate its impact.

2. We convert the platoon dynamics into a state-space model, and apply an ASEKF combined with a detector to smooth the sensor measurement as well as to detect sensor anomalies. For the detector side, we consider both the χ^2-detector and OCSVM. An augmented state formulation is considered in order to compensate for the bias in the state-transition model, which can be caused by stochastic time delay or model inaccuracy. We show using numerical experiments that OCSVM outperforms the χ^2-detector both with and without the augmented state formulation. As we demonstrate in the experiments, however, OCSVM does not necessarily benefit from the augmented state formulation. Experiments also demonstrate that we obtain a significant improvement in detection performance by combining the ASEKF model with a χ^2-detector compared with χ^2-detector without augmented state formulation.
3. One of main advantages of forming platoons is their capability to maintain string stability. Therefore, we conduct a comprehensive stability analysis of platoons under various types of cyberattacks. We define the concept of pseudo string stability to capture the degree of string stability in expectation given an imperfect fault detector. To the best of our knowledge, this is the first string stability analysis of platoons under cyberattacks and model uncertainty. We demonstrate the relationship between the detection sensitivity of the detector and the platoon pseudo string stability, and identify the critical detection sensitivity for maintaining a pseudo stable string.

The remaining of this chapter is organized as follows: In Section 4.2, we review the literature on platooning technology and its cybersecurity concerns. In Section 4.3, we provide the details of ASEKF and detection models, and conduct stability analysis of the platoon with and without attacks. In Section 4.4, we perform extensive numerical experiments. Lastly, we conclude the chapter in Section 4.5.

4.2 Literature Review

CAVs can provide safety, mobility, and sustainability benefits by enabling the formation of platoon. Platoons can increase road capacity, improve traffic stability, and curb fuel consumption (Van Arem et al., 2006; Ploeg et al., 2011; Liu et al., 2021, 2022). A CAV platoon is enabled by using V2V

and/or V2I technologies. It has been demonstrated that the destabilization effect of communication delay is suppressed by the stabilization effect of multi-anticipations (i.e. more cooperative vehicles) of platoon (Ngoduy, 2015). Platooning can also reduce fuel consumption, which is significantly influenced by air resistance, through shorter following gaps. It has been shown that tightly coupled platooning can improve fuel economy for both passenger cars (Liu et al., 2021; Shida & Nemoto, 2009) and trucks (Alam, 2011; Sun, 2020). One application of platooning is cooperative adaptive cruise control (CACC). CACC extends the adaptive cruise control (ACC) by utilizing the V2V communication technology, which provides the ACC system with more and higher-quality information about its immediate following vehicle. With information of this type, the CACC controller will be able to better anticipate challenges ahead, and maneuver in a safer way while at the same time making the ride more comfortable for vehicle occupants. CACC systems with V2V communications enable a reduction in the mean following time gap/headway from about 1.4 seconds in manual driving to approximately 0.6 seconds (Nowakowski et al., 2010).

There has been a considerable amount of literature over the past few years on network-aware modeling of a platoons and improving the string stability of platoon-based vehicular systems. For instance, the effects of communication delays on string stability of a longitudinal dynamic model is addressed in (Zhang & Orosz, 2016; Sykora et al., 2020). Longitudinal platoon control via communication channels with packet loss is studied in (Guo & Wen, 2015; Molnár et al., 2015). In (Molnár et al., 2017), Molnár et al. further investigated the impact of network delay integrated with packet loss on the platoon stability. Although there has been a variety of literature considering network-induced phenomena in the vehicular platoon, there exits much fewer studies considering the detection of cyberattack and their impact on platoon stability. Since the platoon control model heavily relies on the external dynamical information, such as location, velocity, and headway, of other vehicles, when the vehicle communication network is under attack, transmitted messages will be contaminated or lost, rendering the platoon incapable of achieving the expected performance.

Previous literature have considered the impact of cyberattacks on vehicular platoons via simulation (Cui et al., 2018; Khattak et al., 2021), but they did not rigorously investigate the effect of cyberattacks on platoon stability. In (Ngoduy, 2015), a car-following model was designed to receive the velocity and location of a fixed number of cooperative vehicles with constant information transmission time delays. Following this direction, Wang et al. (2020a) extended the framework by incorporating a communication range to dynamically adjust the number of cooperative vesicles. Alipour-Fanid et al. (2017) exemplified a CACC model where an UAV imposed jamming attacks on the wireless channel. Based on the CACC framework, other types of cyberattacks have been studied recently. Mousavinejad et al. (2019) considered the attacks on not only the inter-vehicle signals, but also the onboard sensor measurement outputs. Biron et al. (2018) focused on detecting the Denial of Service (DoS) attack using CACC model and estimating the effect on the CAV system if

attacks occurred. While these studies provided valuable insights on vehicular platoon performance under cyberattacks, the traditional CACC model, which mostly only utilizes one leading vehicle's information for each ego vehicle, cannot fully leverage the Vehicle-to-Everything (V2X) capability, which is key in addressing future challenges in dynamics control and fuel consumption reduction. Therefore, an advanced platoon vehicle dynamics model should be deployed.

To mitigate the risk of being attacked, both detection and recovery techniques are necessary to safeguard platoon operations. Little attention has been paid to cyberattack detection. Mousavinejad et al. (2018) proposed a cyberattack detection algorithm that is capable of detecting attacks that violate both measurements and control command data in platoon-based vehicular systems. Biroon et al. (2021) developed a partial differential equation model for detection and isolation of cyberattacks. They considered a specific type of attack, which is modeled as a ghost vehicle being injected into the connected vehicles network to disrupt the performance of the entire system. In (Ju et al., 2020), the authors proposed a distributed Kalman filter with a modified generalized likelihood ratio algorithm to detect deception attacks. However, all of aforementioned studies only considered a specific platoon dynamic model and a rather specific attack model, which may limit the generalizability of their conclusions in practice. Moreover, none of them considered the time delay effect. This chapter aims to bridge this gap by proposing a general framework describing platoon dynamic, which considers the heterogeneous time delay effect as well as multiple types of sensor anomalies resulted from either sensor faults or cyberattacks.

4.3 Solution Methodology

In this section, we first propose a general model to describe the longitudinal dynamics of CAVs in a platoon. Then, we discuss how to reconfigure the platooning model to a state-space model, which includes a continuous state-transition model with heterogeneous time delays in an augmented state formulation, and a discrete measurement model. The continuous state-transition model represents the intrinsic nature of vehicle motion, and the discrete measurement model represents the discrete nature of sensor sampling. Based on the derived state-space model, we propose a filtering and anomaly detection method, which combines ASEKF with an anomaly detector. For the anomaly detector, we adopt a semi-supervised learning model, namely, OCSVM. We also introduce a χ^2-detector and later compare its performance with OCSVM in Section 4.4. Finally, we conduct string stability analysis of the proposed platooning model under cyberattacks, and propose a novel definition of string stability under cyberattacks and model uncertainty.

4.3.1 Platooning Vehicle Dynamics with Heterogeneous Time Delay

We consider an extended version of the platoon dynamics model proposed by Wang et al. (Wang et al., 2020a). Specifically, the CAV platoon dynamics in the absence of cyberattacks can be modeled as follows:

$$\begin{aligned}\dot{v}_n(t) = f\Bigg(& v_n(t-\tau_1), \alpha_{n1} w_{n1}(t-\tau_1) g_n(t-\tau_1) + \sum_{j=2}^{M} \alpha_{nj} w_{nj}(t-\tau_2) g_{n-j+1}(t-\tau_2), \\ & \beta_{n1} w_{n1}(t-\tau_1) \Delta v_n(t-\tau_1) + \sum_{j=2}^{M} \beta_{nj} w_{nj}(t-\tau_2) \Delta v_{n-j+1}(t-\tau_2) \Bigg)\end{aligned} \tag{4.1}$$

where $v_n(t)$ represents the velocity of n-th vehicle; $x_n(t)$ is the location of the n-th vehicle; $g_n(t)$ represents the clearance gap between the n-th vehicle and $(n-1)$-th vehicle, which is defined as $g_n(t) := x_{n-1}(t) - x_n(t) - l_{n-1}$; l_{n-1} represents the length of $(n-1)$-th vehicle; $\Delta v_n(t)$ is the relative velocity between n-th vehicle and $(n-1)$-th vehicle defined as $\Delta v_n(t) := v_n(t) - v_{n-1}(t)$; α_{nj} and β_{nj} represent the weighting coefficients associated with the clearance gap and relative velocity between vehicle pair $n-j$ and $n-j+1$; M represents the number of cooperative leading vehicles; τ_1 and τ_2 represents the time delay of onboard measurement and communication channel, respectively; $w_{nj}(t)$ is the row-n-column-$(n-j+1)$ element of the adjacency matrix $\mathbf{W}(t)$, and $w_{nj}(t) = 1$ if vehicle n receives information from vehicle $n-j+1$ at time t and $w_{nj}(t) = 0$ otherwise. For example, considering a platoon of N vehicles indexed from 0 to $N-1$, when assuming each vehicle in the platoon only receives information from its leading/preceding vehicles, the adjacency matrix $\mathbf{W}(t)$ will be a lower triangular matrix and can be represented as:

$$\mathbf{W}(t) = \begin{bmatrix} 0 & 0 & \cdots & 0 & \cdots & 0 & 0 \\ w_{11}(t) & 0 & \cdots & 0 & \cdots & 0 & 0 \\ w_{21}(t) & w_{22}(t) & 0 & \ddots & 0 & 0 & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots \\ w_{n1}(t) & w_{n2}(t) & \cdots & w_{nn}(t) & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \ddots & \vdots \\ w_{(N-1)1}(t) & w_{(N-1)2}(t) & \cdots & w_{(N-1)j}(t) & \cdots & w_{(N-1)(N-1)}(t) & 0 \end{bmatrix} \tag{4.2}$$

By defining the n-th vehicle as the ego vehicle, the platooning model in equation (4.1) takes three inputs, namely, velocity of the ego vehicle, weighted average of clearance gaps, and weighted average of relative velocities between each pair of its cooperative vehicles. The platooning model

determines the acceleration of the ego vehicle. Note that we consider two heterogeneous time delay factors, one for onboard measurements and one for the communication channel. To be specific, a time delay factor τ_1 incurs on all onboard measurements of the ego vehicle, including its velocity, the clearance gap and the relative velocity between the ego vehicle and its immediate leading vehicle. Meanwhile, another time delay factor τ_2 is imposed to the rest of the inputs of model (4.1), which are obtained via the communication channel. For a complete list of notations, see Table 4.1.

Table 4.1: Table of Notation

$f(\cdot)$	$\triangleq$	general platooning model
N	$\triangleq$	number of vehicles in platoon
$\mathbf{W}(t)$	$\triangleq$	adjacency matrix at time t
τ_1	$\triangleq$	time delay of onboard measurements
τ_2	$\triangleq$	time delay of the communication channel
l_n	$\triangleq$	length of vehicle n
$v_n(t)$	$\triangleq$	velocity of vehicle n in the platoon at time t
$x_n(t)$	$\triangleq$	position of vehicle n in the platoon at time t
$g_n(t)$	$\triangleq$	clearance gap between the preceding vehicle $n-1$ and vehicle n at time t
$\Delta v_n(t)$	$\triangleq$	relative velocity between vehicle n and its preceding vehicle $n-1$ at time t
α_{nj}/β_{nj}	$\triangleq$	weighting coefficients associated with the clearance gap/relative velocity from vehicle $n-j$ to vehicle $n-j+1$

4.3.2 State-Space Model

We define a state-space model which includes a continuous state-transition model and a discrete measurement model. Define the state vector $s_n(t)$ as the location and the velocity of the ego vehicle, i.e.,

$$s_n(t) = [x_n(t), v_n(t)]^\intercal \tag{4.3}$$

where $\intercal$ represents transpose. Also define the input vector as $u_n(t; \tau_1, \tau_2)$, which associates with two time delay factors τ_1 and τ_2 and includes the clearance gap and the relative velocity,

$$u_n(t;\tau_1,\tau_2) = \begin{bmatrix} \alpha_{n1} w_{n1} g_n(t-\tau_1) + \sum_{j=2}^{M} \alpha_{nj} w_{nj}(t-\tau_2) g_{n-j+1}(t-\tau_2) \\ \beta_{n1} w_{n1}(t-\tau_1)\Delta v_n(t-\tau_1) + \sum_{j=2}^{M} \beta_{nj} w_{nj}(t-\tau_2)\Delta v_{n-j+1}(t-\tau_2) \end{bmatrix} \tag{4.4}$$

The platoon dynamics (4.1) can be therefore recast into a state-transition model as follows,

$$\begin{aligned}\dot{s}_n(t) &= \begin{bmatrix}\dot{x}_n(t)\\ \dot{v}_n(t)\end{bmatrix}\\ &= \begin{bmatrix} e_2^\intercal s_n(t) \\ f\left(e_2^\intercal s_n(t-\tau_1), e_1^\intercal u_n(t|\tau_1,\tau_2), e_2^\intercal u_n(t|\tau_1,\tau_2)\right)\end{bmatrix}\end{aligned} \tag{4.5}$$

where e_i represents a base vector with its i-th element being 1, and other elements set to 0.

When $\tau_1 = 0$, the state-transition model (4.5) satisfies the Markovian property, allowing for applying an extended Kalman filter (EKF). However, because of the time required for data processing and computations, in practice τ_1 can be non-zero. Similarly, communication delay may cause τ_2 to be non-zero. Under such circumstances, the Kalman filer cannot be directly applied to equation (4.5). Instead, we approximate the state-transition model by using an augmented state formulation. Specifically, assuming each vehicle has a bounded acceleration range, we can obtain a delay differential equation (DDE), describing the delayed state-transition model:

$$\begin{aligned}\dot{s}_n(t) &= \begin{bmatrix}\dot{x}_n(t)\\ \dot{v}_n(t)\end{bmatrix}\\ &= \begin{bmatrix} e_2^\intercal s_n(t-\tau_1) + \int_{t-\tau_1}^{t} a_n(r)dr \\ f(e_2^\intercal s_n(t-\tau_1), e_1^\intercal u_n(t|\tau_1,\tau_2), e_2^\intercal u_n(t|\tau_1,\tau_2))\end{bmatrix}\\ &= \begin{bmatrix}1,0,1\\ 0,1,0\end{bmatrix} \times \begin{bmatrix} e_2^\intercal s_n(t-\tau_1) \\ f(e_2^\intercal s_n(t-\tau_1), e_1^\intercal u_n(t|\tau_1,\tau_2), e_2^\intercal u_n(t|\tau_1,\tau_2)) \\ \int_{t-\tau_1}^{t} a_n(r)dr\end{bmatrix}\\ &= \begin{bmatrix}1,0,1\\ 0,1,0\end{bmatrix} \times \begin{bmatrix} e_2^\intercal \tilde{s}_n(t-\tau_1) \\ f(e_2^\intercal \tilde{s}_n(t-\tau_1), e_1^\intercal u_n(t|\tau_1,\tau_2), e_2^\intercal u_n(t|\tau_1,\tau_2)) \\ e_3^\intercal \tilde{s}_n(t-\tau_1)\end{bmatrix}\end{aligned} \tag{4.6}$$

where we define the augmented state vector as

$$\tilde{s}_n(t) = [x_n(t), v_n(t), \delta_n(t)]^\intercal \tag{4.7}$$

with augmented state

$$\delta_n(t-\tau) = \begin{cases}\int_{t-\tau}^{t} a_n(r)dr & \text{if } \tau > 0\\ 0 & \text{otherwise}\end{cases} \tag{4.8}$$

The augmented state $\delta_n(t)$ is used to compensate potential bias caused by the time delay. Since $\delta_n(t)$ is unknown, we assume that at each time it is sampled from a random process with a small

variance. Thus we have $\dot{\delta}_n(t) \approx 0$.

Then we can obtain the state-transition model with the augmented state vector $\tilde{s}_n(t)$ as follows:

$$\begin{aligned}\dot{\tilde{s}}_n(t) &= \begin{bmatrix}\dot{x}_n(t)\\ \dot{v}_n(t)\\ \dot{\delta}_n(t)\end{bmatrix}\\ &\approx \begin{bmatrix} e_2^\intercal \tilde{s}_n(t-\tau_1)\\ f(e_2^\intercal \tilde{s}_n(t-\tau_1), e_1^\intercal u_n(t;\tau_1,\tau_2), e_2^\intercal u_n(t;\tau_1,\tau_2))\\ 0\end{bmatrix} + \theta(t)\\ &= \mathcal{T}(\tilde{s}_n(t-\tau_1), u_n(t;\tau_1,\tau_2)) + \theta(t)\end{aligned} \tag{4.9}$$

where $\mathcal{T}(\cdot)$ is the state-transition model, and $\theta(t)$ is the process noise which accounts for the approximation error and model inaccuracy.

At each time epoch, the ego vehicle obtains its trajectory information from measurements by its onboard sensors. As such, using the new augmented state vector $\tilde{s}_n(t)$, we can obtain the state-space model with an augmented state vector as follows:

$$\begin{aligned}\dot{\tilde{s}}_n(t) &= \mathcal{T}(\tilde{s}_n(t-\tau_1), u_n(t;\tau_1,\tau_2)) + \theta(t)\\ z_n(t_k) &= \mathcal{M}(\tilde{s}_n(t_k)) + \eta(t_k),\ k \in \{0 \cup \mathbb{Z}^+\}\end{aligned} \tag{4.10}$$

where $\mathcal{M}(\cdot)$ represents the measurement model, $z_n(\cdot)$ is sensor readings of the ego vehicle, $\eta(t_k)$ denotes the observation noise, which is assumed to be mutually independent from the process noise, $t_{k+1} = t_k + \Delta t,\ k \in \{0 \cup \mathbb{Z}^+\}$, and Δt is the sampling time interval for sensors.

4.3.3 Stochastic Time Delay of Input

We further consider a more general setting where the time delay factors τ_1 and τ_2 are not known constants, i.e., the input vector suffers from stochastic time delay. We assume the stochastic time delay obeys a linear model,

$$\begin{aligned}\tilde{\tau}_1 &= \tau_1 + \kappa_1\\ \tilde{\tau}_2 &= \tau_2 + \kappa_2\end{aligned} \tag{4.11}$$

where κ_1 and κ_2 follow truncated normal distributions with mean 0 and variance σ_1^2 and σ_2^2, and within the intervals (a_1, b_1) and (a_2, b_2), respectively. That is, time delays $\tilde{\tau}_1$ and $\tilde{\tau}_2$ are within the range $(\tau_1 + a_1, \tau_1 + b_1)$ and $(\tau_2 + a_2, \tau_2 + b_2)$, respectively. The dynamics of the ego vehicle's most

immediate leader can be simplified as a linear model,

$$\begin{cases} \dot{x}_{n-1}(t) = v_{n-1}(t) \\ \dot{v}_{n-1}(t) = a_{n-1}(t) \end{cases} \tag{4.12}$$

where $a_{n-1}(t)$ is the acceleration of the $n-1$-th vehicle at time t. Then, we can obtain the following proposition:

Proposition 4.1. *Having stochastic time delays $\tilde{\tau}_1$ and $\tilde{\tau}_2$ is equivalent to adding noises into the input vector $u_n(t;\tau_1,\tau_2)$ with fixed time delays τ_1 and τ_2, i.e.,*

$$u_n(t;\tilde{\tau}_1,\tilde{\tau}_2) = u_n(t;\tau_1,\tau_2) + C(t) \tag{4.13}$$

where $C(t)$ represents the noises caused by stochastic time delay.

Proof. Let us start by considering the communication delay, τ_2. For an arbitrary cooperative leader of the ego vehicle, i.e. $(n-j)$-th vehicle where $1 \leq j \leq M$, its clearance gap $g_{n-j}(t-\tilde{\tau}_2)$ is defined as $g_{n-j}(t-\tilde{\tau}_2) = x_{n-j-1}(t-\tilde{\tau}_2) - x_{n-j}(t-\tilde{\tau}_2) - l_{n-j-1}$. By integrating $\dot{x}_{n-j-1}(t-\tilde{\tau}_2)$, we have

$$\begin{aligned} x_{n-j-1}(t-\tilde{\tau}_2) &= \int_0^{t-\tilde{\tau}_2} v_{n-j-1}(\xi)d\xi \\ &= \int_0^{t-\tau_2-\kappa_2} v_{n-j-1}(\xi)d\xi \\ &= \int_0^{t-\tau_2} v_{n-j-1}(\xi)d\xi - \int_{t-\tau_2-\kappa_2}^{t-\tau_2} v_{n-j-1}(\xi)d\xi \\ &= x_{n-j-1}(t-\tau_2) + \epsilon_1(t-\tau_2;\kappa_2) \end{aligned} \tag{4.14}$$

where $\epsilon_1(t-\tau_2;\kappa_2) = -\int_{t-\tau_2-\kappa_2}^{t-\tau_2} v_{n-j-1}(\xi)d\xi$. Similarly, by integrating $x_{n-j}(t-\tilde{\tau}_2)$, we can obtain,

$$x_{n-j}(t-\tilde{\tau}_2) = x_{n-j}(t-\tau_2) + \epsilon_2(t-\tau_2;\kappa_2) \tag{4.15}$$

where $\epsilon_2(t-\tau_2;\kappa_2) = -\int_{t-\tau_2-\kappa_2}^{t-\tau_2} v_{n-j}(\xi)d\xi$.

Therefore, by combing (4.14) and (4.15), we obtain

$$\begin{aligned} g_{n-j}(t-\tilde{\tau}_2) &= x_{n-j-1}(t-\tau_2) - x_{n-j}(t-\tau_2) - l_{n-j-1} + \epsilon_1(t-\tau_2;\kappa_2) - \epsilon_2(t-\tau_2;\kappa_2) \\ &= g_{n-j}(t-\tau_2) + \epsilon_1(t-\tau_2;\kappa_2) - \epsilon_2(t-\tau_2;\kappa_2) \end{aligned} \tag{4.16}$$

Equation (4.16) shows that having stochastic time delay on the clearance gap is equivalent to adding a noise term $\epsilon_1(t-\tau_2;\kappa_2) - \epsilon_2(t-\tau_2;\kappa_2)$ into the clearance gap with a constant time delay. Note that we only consider the communication delay τ_2. However, similar results can be easily

obtained for onboard measurement delay τ_1. Also note that the noise term is not necessarily zero mean with respect to κ_2, since it also depends on the vehicle velocity:

$$\begin{aligned}
&\mathbb{E}_{\kappa_2}[\epsilon_1(t-\tau_2;\kappa_2)-\epsilon_2(t-\tau_2;\kappa_2)] = \mathbb{E}_{\kappa_2}[\epsilon_1(t-\tau_2;\kappa_2)] - \mathbb{E}_{\kappa_2}[\epsilon_2(t-\tau_2;\kappa_2)] \\
&= \int_{a_2}^{b_2} \tilde{\phi}(\iota;0,\sigma_2,a_2,b_2) \int_0^{-\iota} v_{n-j-1}(\xi+t-\tau_2)d\xi d\iota \\
&\quad - \int_{a_2}^{b_2} \tilde{\phi}(\iota;0,\sigma_2,a_2,b_2) \int_0^{-\iota} v_{n-j}(\xi+t-\tau_2)d\xi d\iota \\
&= \int_{a_2}^{b_2} \tilde{\phi}(\iota;0,\sigma_2,a_2,b_2) \int_0^{-\iota} \Delta v_{n-j}(\xi+t-\tau_2)d\xi d\iota
\end{aligned} \tag{4.17}$$

where $\tilde{\phi}(\iota;\mu,\sigma,a,b)$ represents the probability density function of truncated normal distribution,

$$\tilde{\phi}(\iota;\mu,\sigma,a,b) = \frac{1}{\sigma}\frac{\phi(\frac{\iota-\mu}{\sigma})}{\Phi(\frac{b-\mu}{\sigma})-\Phi(\frac{a-\mu}{\sigma})} \tag{4.18}$$

where $\phi(\cdot)$ and $\Phi(\cdot)$ are the probability density function and cumulative density function of the standard normal distribution, respectively.

Similarly, by integrating $v_{n-j-1}(t-\tilde{\tau}_2)$ and $v_{n-j}(t-\tilde{\tau}_2)$, we obtain

$$\begin{aligned}
v_{n-j-1}(t-\tilde{\tau}_2) &= v_{n-j-1}(t-\tau_2) + \epsilon_3(t-\tau_2;\kappa_2) \\
v_{n-j}(t-\tilde{\tau}_2) &= v_{n-j}(t-\tau_2) + \epsilon_4(t-\tau_2;\kappa_2)
\end{aligned} \tag{4.19}$$

where $\epsilon_3(t-\tau_2;\kappa_2) = -\int_{t-\tau_2-\kappa_2}^{t-\tau_2} a_{n-j-1}(\xi)d\xi$ and $\epsilon_3(t-\tau_2;\kappa_2) = -\int_{t-\tau_2-\kappa_2}^{t-\tau_2} a_{n-j}(\xi)d\xi$. Then we have

$$\Delta v_{n-j}(t-\tilde{\tau}_2) = \Delta v_{n-j}(t-\tau_2) + \epsilon_3(t-\tau_2;\kappa_2) - \epsilon_4(t-\tau_2;\kappa_2) \tag{4.20}$$

Therefore according to (4.16) and (4.20), having stochastic time delays is equivalent to adding noises into the input vector with the fixed time delays. □

When the time delays of the input vector are stochastic, according to Proposition 4.1, substituting equation (4.13) into (4.10) could induce a non-zero mean for the process noise $\theta(t)$, depending on the specific formulation of the platooning model. As mentioned in the next section, such a bias in $\theta(t)$ could negatively affect the performance of the classic χ^2-detector, whereas using ASEKF and OCSVM can mitigate this issue.

4.3.4 Augmented State Extended Kalman Filter with Anomaly Detector

Extended Kalman filter (EKF) is a well-known algorithm that takes a series of observed measurements and estimates the unknown state of a non-linear system in a timely and accurate manner (Ribeiro, 2004). However, similar to other types of Kalman filter-based algorithms, it poses an assumption on both the process noise and the observation noise to be zero-mean Gaussian distributed. It has been shown that regardless of the Gaussian assumption, if the process covariance and measurement covariance are known, the Kalman filter is still the best possible linear estimator in the sense of minimum mean-squared-error (Humpherys et al., 2012). However, its performance can deteriorate significantly when there exits a background bias that is not incorporated in the model, as it violates the zero-mean assumption. In order to denoise CAV sensor measurements while compensating potential but unknown biases, we apply an augmented state extended Kalman filter (ASEKF) to the state-space model in (4.10) with three objectives: (i) to smooth the CAV sensor noise and estimate vehicle state in real time, (ii) to compensate potential but unknown bias caused by stochastic time delays or model inaccuracy, and (iii) to detect anomalous sensor readings by incorporating surrounding vehicles' information.

ASEKF includes two major stages to obtain the state estimation of $\tilde{s}_n(t_k)$ from the sensor input $z_n(t_k)$, namely, predict and update. Let $\hat{s}(k|k-1)$ and $P(t_k|t_{k-1})$ denote the state prediction and state covariance prediction at time t_k given the estimate at time t_{k-1}, respectively. Note that for ease of notation, we use state vector notation s instead of the augmented state vector $\tilde{s}$, and we also omit subscript n for simplicity. Hence, given the state-space model in equation (4.10), we can adopt the EKF algorithm introduced in Chapter 2.3.

One advantage of using ASEKF to estimate sensor data is that it can detect anomalies during the filtering procedure. One of the traditional anomaly detectors used in conjunction with Kalman filter is the χ^2-detector (Brumback & Srinath, 1987; Bar-Shalom & Li, 1995). Since ASEKF belongs to the family of Kalman filters, the χ^2-detector can be seamlessly applied. Specifically, it constructs a gate region by computing the χ^2 test statistics, and determines whether the new measurement falls into the gate region. The gate region is defined by the gate threshold γ, as shown in the following:

$$V_\gamma(k) = \{z : (z - \hat{z}_{k|k-1})^\intercal S_k^{-1} (z - \hat{z}_{k|k-1}) \leq \gamma\} \tag{4.21}$$

where $\hat{z}_{k|k-1}$ is the predicted value of measurement at time t_k. The χ^2 test statistics for the anomaly detector is defined as

$$\chi^2(t_k) = \nu_k^\intercal S_k^{-1} \nu_k \tag{4.22}$$

The χ^2-detector relies on the Gaussian assumption and zero-mean assumption of the ASEKF, as it essentially constructs a "spherical" decision boundary with the centroid of the origin point in the

space of normalized innovation, which is defined as

$$\bar{\nu}(t_k) = S_k^{-\frac{1}{2}} \cdot \nu_k \tag{4.23}$$

The normalized innovation instances falling outside this spherical decision boundary will be classified as anomalies. However, as we showed earlier in section 4.3.3, the process noise $\theta(t)$ may not be zero-mean under stochastic time delay, and the additive noise caused by the stochastic time delay does not follow a zero-mean Gaussian distribution. Moreover, the approximation step in (4.9) may also introduce such a bias. Therefore, we also consider a learning-based method, namely, one class Support Vector Machine (OCSVM), to actively learn the decision boundary in the normalized innovation space. To see the details of OCSVM, refer to model (2.15) in Chapter 2.

Unlike the χ^2-detector which uses a decision boundary predefined by the threshold parameter γ, OCSVM learns the decision boundary from only non-anomalous training data, which can be collected easily without the need to enumerate all possible types of anomalies. It can also directly learn the potential bias from the training data, and is more robust. Furthermore, unlike the χ^2-detector, it does not impose distributional assumptions on the data.

4.3.5 String Stability Analysis

String stability reflects how platoons respond to imposed perturbations in longitudinal dynamics and stabilize back to the equilibrium state. String stability can be mathematically defined in both time-domain and frequency domain. String stability conditions are easy to verify, but hard to derive, in time-domain. Therefore, we conduct analysis in the frequency-domain. Note that in the presence of time delay, deploying power-series and the decay rate of perturbations to approximate the vehicle dynamics response and derive string stability conditions is not mathematically guaranteed for a non-linear dynamics model at high angular frequency. To ensure the validity of the analysis, we pursue the transfer function approach instead. Obtaining inter-vehicle transfer functions under the standard definition of string stability is not trivial when their inputs and outputs are heavily correlated, i.e., one vehicle receiving information from multiple preceding vehicles as its control input. In this chapter, we adopt an extension of the standard string stability, namely, the head-to-tail string stability, originally proposed in (Jin & Orosz, 2014) to address this problem:

Definition 4.1. *A platoon is called head-to-tail string stable if any perturbations that cause the first vehicle in the platoon (i.e., the platoon head) to deviate from its equilibrium state can be attenuated at the very last vehicle (i.e. the platoon tail).*

Head-to-tail string stability views a platoon of any size as a system with input (perturbation at the platoon head) and output (perturbation at the platoon tail). This input-output relationship

between the platoon head and any vehicle following it can be established by truncating the platoon at the corresponding number of vehicles. This facilitates describing vehicle responses in complicated longitudinal dynamics, i.e., the vehicle takes outputs of multiple preceding vehicles, and its output serves as an input for other vehicles. Specifically, a transfer function that connects the input of the platoon head and the output of the platoon tail is called a head-to-tail transfer function, as described in (Jin & Orosz, 2014). For a subject vehicle in the platoon, its correlated inputs can be neatly decoupled and represented in the form of head-to-tail transfer functions, using multiple transfer functions of any two vehicles between the platoon head and the subject vehicle itself. A detailed description will be provided later in this subsection. To better understand head-to-tail string stability and how it works with control dynamics, an example of vehicular communication topology is shown in Figure 4.2.

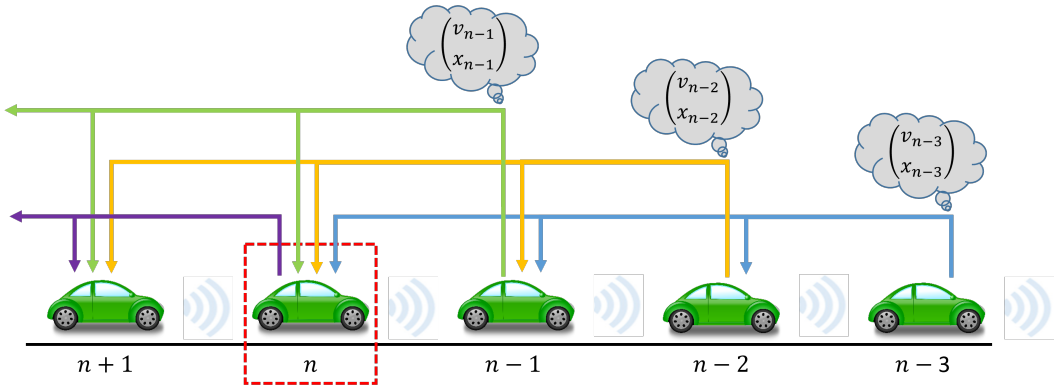


Figure 4.2: The topology in the above figure is called M-predecessor following (MPF), where here M=3 denotes the number of predecessors of vehicle n with communication capabilities. Vehicles n to $n-3$ are called a cooperative vehicle group. State information of position and velocity are transmitted via the vehicular network.

With the state space model defined earlier, the state of vehicle n at time t under flow equilibrium conditions is denoted as $s_n^*(t) = [x_n^*(t), v_n^*(t)]^\intercal$, where $x_n^*(t) = x_n(0) + t \cdot v_n^*(t)$ is the expected position of vehicle n at time t without perturbations. The actual position of vehicle n at time t is denoted as $x_n(t)$, and the vehicle length is l. One can easily obtain $g^* = x_{n-1}^*(t) - x_n^*(t) - l = x_{n-1}^*(0) - x_n^*(0) - l$, which is a constant determined by the initial condition of vehicle n. When perturbations are imposed at time t, the relationship between the perceived position and velocity, the actual position and velocity, and perturbations can be formulated as follows:

$$\begin{aligned} \tilde{x}_n(t) &= x_n^*(t) - x_n(t) \\ \tilde{v}_n(t) &= v_n^*(t) - v_n(t) \end{aligned} \tag{4.24}$$

where $\tilde{x}_n(t)$ and $\tilde{v}_n(t)$ are the perturbations imposed on location and velocity, respectively. For

vehicle n which utilizes information received from its cooperative leading vehicles, its longitudinal dynamics model can be linearized as follows:

$$\begin{aligned}\dot{\tilde{v}}_n(t) = f_n^v \tilde{v}_n(t-\tau_1) + f_n^g\Big(\alpha_{n1} w_{n1}(t-\tau_1)\tilde{g}_n(t-\tau_1) + \sum_{j=2}^{M} \alpha_{nj} w_{nj}(t-\tau_2)\tilde{g}_{n-j+1}(t-\tau_2)\Big) \\ + f_n^{\Delta v}\Big(\beta_{n1} w_{n1}(t-\tau_1)\Delta\tilde{v}_n(t-\tau_1) + \sum_{j=2}^{M} \beta_{nj} w_{nj}(t-\tau_2)\Delta\tilde{v}_{n-j+1}(t-\tau_2)\Big)\end{aligned} \tag{4.25}$$

where

$$f_n^v = \left.\frac{\partial f}{\partial \tilde{v}_n}\right|_{s=s_n^*},\ f_n^g = \left.\frac{\partial f}{\partial \tilde{g}_n}\right|_{s=s_n^*}, f_n^{\Delta v} = \left.\frac{\partial f}{\partial \Delta\tilde{v}_n}\right|_{s=s_n^*} \tag{4.26}$$

The adjacency matrix $\mathbf{W}$ is omitted since the communication topology is fixed for M leading vehicles. We also denote α_{nj} and β_{nj} as α_j and β_j, respectively, for simplicity. The corresponding state space model can be formulated as:

$$\begin{aligned}\dot{\tilde{s}}_n(t) &= \begin{bmatrix}\dot{\tilde{x}}_n(t)\\ \dot{\tilde{v}}_n(t)\end{bmatrix} \\ &= \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\cdot\begin{bmatrix}\tilde{x}_n(t)\\ \tilde{v}_n(t)\end{bmatrix} + \begin{bmatrix}0 & 0\\ -\alpha_1 f_n^g & f_n^v + \beta_1 f_n^{\Delta v}\end{bmatrix}\cdot\begin{bmatrix}\tilde{x}_n(t-\tau_1)\\ \tilde{v}_n(t-\tau_1)\end{bmatrix} \\ &\quad + \sum_{j=1}^{M-1}\begin{bmatrix}0 & 0\\ (\alpha_j - \alpha_{j+1}) f_n^g & (\beta_{j+1} - \beta_j) f_n^{\Delta v}\end{bmatrix}\cdot\begin{bmatrix}\tilde{x}_{n-j}(t-\tau_2)\\ \tilde{v}_{n-j}(t-\tau_2)\end{bmatrix} \\ &\quad + \begin{bmatrix}0 & 0\\ \alpha_M f_n^g & -\beta_M f_n^{\Delta v}\end{bmatrix}\cdot\begin{bmatrix}\tilde{x}_{n-M}(t-\tau_2)\\ \tilde{v}_{n-M}(t-\tau_2)\end{bmatrix} \\ y_n(t) &= \begin{bmatrix}0 & 1\end{bmatrix}\cdot\begin{bmatrix}\tilde{x}_n(t)\\ \tilde{v}_n(t)\end{bmatrix}\end{aligned} \tag{4.27}$$

After Laplace transformation of equation (4.27), the relationship between the output of vehicle n and its cooperative leading vehicles is shown as follows:

$$\begin{aligned}Y_n(s) &= \begin{bmatrix}0 & 1\end{bmatrix}\cdot\Big(sI - \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix} - A_n^{\tau_1}\cdot e^{-s\tau_1}\Big)^{-1}\cdot\left[\Big(\sum_{j=1}^{M} B_{n-j}^{\tau_2} Y_{n-j}(s)\Big)\cdot e^{-s\tau_2}\begin{bmatrix}\frac{1}{s}\\ 1\end{bmatrix}\right] \\ &= \sum_{j=1}^{M} T_{n-j}(s) Y_{n-j}(s)\end{aligned} \tag{4.28}$$

where

$$
\begin{aligned}
A_n^{\tau_1} &= \begin{bmatrix} 0 & 0 \\ -\alpha_1 f_n^g & f_n^v + \beta_1 f_n^{\Delta v} \end{bmatrix} \\
B_{n-j}^{\tau_2} &= \begin{bmatrix} 0 & 0 \\ (\alpha_j - \alpha_{j+1}) f_n^g & (\beta_{j+1} - \beta_j) f_n^{\Delta v} \end{bmatrix}, \ 1 \leq j \leq M-1 \\
B_{n-M}^{\tau_2} &= \begin{bmatrix} 0 & 0 \\ \alpha_M f_n^g & -\beta_M f_n^{\Delta v} \end{bmatrix}
\end{aligned} \tag{4.29}
$$

Here $T_{n-j}(s)$ represents the transfer function between vehicle $n-j$ and vehicle n. According to the definition, the head-to-tail transfer function is in the form of:

$$
Y_n(s) = G_{n,0}(s) Y_0(s) \tag{4.30}
$$

However, in this case, the head-to-tail transfer function is difficult to derive directly as the outputs of the leading vehicles are highly coupled within one vehicle group (vehicle n and its M cooperative leading vehicles). Inspired by the method proposed in (Zhang & Orosz, 2016), we can derive the head-to-tail transfer function by iteration. By substituting equation (4.30) into equation (4.28), we can get:

$$
G_{n,0}(s) = \sum_{j=1}^{M} T_{n-j}(s) G_{n-j,0}(s) \tag{4.31}
$$

Rearrange the equation (4.31) to a transition model. It can be shown that:

$$
\begin{bmatrix} G_{n,0}(s) \\ G_{n-1,0}(s) \\ G_{n-2,0}(s) \\ G_{n-3,0}(s) \\ \vdots \\ G_{n-M,0}(s) \end{bmatrix} = \begin{bmatrix} T_{n-1}(s) & T_{n-2}(s) & T_{n-3}(s) & \cdots & T_{n-M}(s) & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} G_{n-1,0}(s) \\ G_{n-2,0}(s) \\ G_{n-3,0}(s) \\ G_{n-4,0}(s) \\ \vdots \\ G_{n-M-1,0}(s) \end{bmatrix} \tag{4.32}
$$

From (4.32), for any two sequential vehicle groups, one with starting and ending vehicles $n-M$ and n, respectively, and the other with starting and ending vehicles $n-M-1$ and $n-1$, respectively, the relationship between the two groups of vehicles' head-to-tail transfer functions can be clearly

established. Denote

$$\hat{P}_n(s) = \begin{bmatrix} T_{n-1}(s) & T_{n-2}(s) & T_{n-3}(s) & \cdots & T_{n-M}(s) & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix}$$
$$\mathcal{G}_n(s) = \begin{bmatrix} G_{n,0}(s) \\ G_{n-1,0}(s) \\ G_{n-2,0}(s) \\ G_{n-3,0}(s) \\ \vdots \\ G_{n-M,0}(s) \end{bmatrix} \tag{4.33}$$

where $\hat{P}_n(s)$ is the transfer matrix and $\mathcal{G}_n(s)$ is the vector containing head-to-tail transfer functions of vehicles from n to $n-M$. According to Theorem 2 in (Zhang & Orosz, 2016), if $\|\mathcal{G}_n(s)\| < \|\mathcal{G}_{n-1}(s)\|$ for any two consecutive vehicle groups with leading vehicle n and $n-1$ respectively, then perturbations can be mitigated iteratively from the platoon head to the platoon tail, reaching head-to-tail string stability. We adopt this theorem to a platoon model with identical longitudinal vehicle dynamics. As a result, from equations (4.31)-(4.33), it requires that

$$\sup_{\forall \omega > 0} |\lambda_k(\hat{P}_n(i\omega))| < 1, \quad k = 1, 2, ..., M \tag{4.34}$$

where $\lambda_k(\hat{P}_n(i\omega))$ is the k-th eigenvalue of the transfer matrix $\hat{P}_n(i\omega)$ with frequency ω.

4.3.6 Pseudo String Stability Analysis under Cybersecurity Uncertainties

In section 4.3.5, we conduct string stability analysis of the platooning model (4.1) in the attack-free scenario. In this section, we further extend the stability analysis under cyberattacks while taking the detection and recovery into account. Specifically, we aim to model the system with the ability to detect anomalies and fully recover the true measurements once the anomalies are detected. We assume the recovery can be achieved by utilizing other sources of information, e.g., road side units (RSUs). The detection sensitivity/recall may not always be 100%, meaning that there exists uncertainty in the platooning model where it switches between the compromised model and the normal model. Note that current tools for stability analysis do not consider such probabilistic models. Therefore, in this chapter, we define the concept of *pseudo string stability* for the case

where the model is probabilistic.

We assume all platoon members will be affected by the attack. This can be achieved by a drone or a wireless device conducting false injection attacks or jamming attacks to affect either onboard measurements or the input vector. We also assume each platoon member is equipped with the same detector with detection sensitivity p, which is defined as the number of true positive anomaly detections, divided by the total number of anomalous instances. Then, the platoon model (4.1) becomes a probabilistic model,

$$\begin{aligned}\dot{v}_n(t) = \eta_t f\left(v_n(t-\tau_1), \bar{g}_n(t;\tau_1,\tau_2), \bar{d}_n(t;\tau_1,\tau_2)\right) \\ + (1-\eta_t)\cdot f\left(v_n(t-\tilde{\tau_1}) + \tilde{A}, \bar{g}_n(t;\tilde{\tau}_1,\tilde{\tau}_2) + \tilde{B}, \bar{d}_n(t;\tilde{\tau}_1,\tilde{\tau}_2) + \tilde{C}\right)\end{aligned} \tag{4.35}$$

where

$$\bar{g}_n(t;\tau_1,\tau_2) := \alpha_1 g_n(t-\tau_1) + \sum_{j=2}^{M} \alpha_j g_{n-j+1}(t-\tau_2)$$

$$\bar{d}_n(t;\tau_1,\tau_2) := \beta_1 \Delta v_n(t-\tau_1) + \sum_{j=2}^{M} \beta_j \Delta v_{n-j+1}(t-\tau_2)$$

and η_t is a Bernoulli random variable at time t with $\mathbb{P}(\eta_t = 1) = \tilde{p} = p^N$ and $\mathbb{P}(\eta_t = 0) = 1 - \tilde{p}$ given N vehicles in the platoon. Note that for the ease of stability analysis and for security concerns, we adopt the most conservative setting where if any platoon member fails to detect the attack, the whole platooning model becomes compromised with $\eta = 0$. The attack parameters are $\tilde{\tau}_1$, $\tilde{\tau}_2$, $\tilde{A}$, $\tilde{B}$, and $\tilde{C}$, where $\tilde{\tau}_1$, $\tilde{\tau}_2$ can be affected by jamming attacks and the rest of three parameters can be affected by false injection attacks. Note that without abuse of notation we denote $\tilde{\tau}_1$ and $\tilde{\tau}_2$ as any time delays different from the single values τ_1 and τ_2, which can also account for stochastic time delays.

Since the platoon model in (4.35) is a probabilistic model, we define pseudo string stability under model uncertainty as follows:

Definition 4.2. *Consider a vehicle string with semi-infinite length in equilibrium state. Impose a transient perturbation on the head vehicle. The vehicle string is pseudo string stable if the perturbation eventually vanishes when reaching the tail vehicle in the string.*

Note that by Definition 4.2 the perturbation could be amplified for some time periods and a subset of vehicles. However, if the vehicle string is sufficiently long, the perturbation will vanish at the tail vehicle. Note that this is different from Definition 4.1, which requires perturbation attenuates at the end of the vehicle string even for a finite-length vehicle string.

Denote the transfer matrix of the compromised dynamic model as

$$\hat{P}_n(s;\Lambda) = \begin{bmatrix} T_{n-1}(s;\Lambda) & T_{n-2}(s;\Lambda) & \dots & T_{n-M}(s;\Lambda) & 0 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix} \tag{4.36}$$

where Λ represents the set of attack parameters $\tilde{\tau}_1$, $\tilde{\tau}_2$, $\tilde{A}$, $\tilde{B}$, and $\tilde{C}$. Denote the transfer matrix of the probabilistic platooning model (4.35) as $\hat{P}_n(s;\Lambda,\tilde{p})$. Then, given a detection sensitivity p, we can obtain the mean transfer matrix of the probabilistic platooning model (4.35),

$$\begin{aligned} \bar{\hat{P}}_n(s;\Lambda) &:= \mathbb{E}_{\tilde{p}}\left[\hat{P}_n(s;\Lambda,\tilde{p})\right] \\ &= \tilde{p}\cdot\hat{P}_n(s) + (1-\tilde{p})\cdot\hat{P}_n(s;\Lambda) \\ &= \begin{bmatrix} \bar{T}_{n-1}(s;\Lambda) & \bar{T}_{n-2}(s;\Lambda) & \dots & \bar{T}_{n-M}(s;\Lambda) & 0 \\ 1 & 0 & \dots & 0 & 0 \\ 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \dots & 1 & 0 \end{bmatrix} \end{aligned} \tag{4.37}$$

where $\bar{T}_i(s;\Lambda) = \tilde{p}\cdot T_i(s) + (1-\tilde{p})\cdot T_i(s;\Lambda)$.

Given a stochastic model $\mathcal{F}(\tilde{p},\hat{\xi})$, it is pseudo string stable if it satisfies

$$\sup_{\forall\omega>0}\left|\lambda_k(\bar{\hat{P}}_n(i\omega;\Lambda))\right| < 1, \quad k = 1,2,...,M \tag{4.38}$$

By formulas (4.37) and (4.38), we can validate whether a detection sensitivity can ensure a pseudo stable string given a set of attack parameters Λ.

After showing that the head-to-tail string stability is a function of detection sensitivity, as illustrated later in Section 4.4, it is worthwhile and possible to find the critical detection sensitivity under a set of specific cyberattack parameters, such that if all detectors can successfully detect and recover from the attacks with a probability higher than the critical detection sensitivity, then the pseudo head-to-tail string stability can be maintained.

4.4 Numerical Experiment

In this section, we perform extensive numerical experiments to investigate the anomaly detection performance of our proposed methods in Section 4.3. First, we investigate the performance of the χ^2-detector and OCSVM under a mixed set of anomaly types introduced in Section 4.1, namely, 'short', 'noise', 'bias', 'gradual drift', and 'miss', with random attack magnitude and duration. This experiment explores the potential of using OCSVM and an augmented state formulation in the presence of sensor measurement and communication time delays. Next, we conduct sensitivity analysis on the attack parameters and investigate their impact on platoon string stability. This experiment demonstrates the stability of the platoon under different combinations of attack scenarios. Lastly, we analyze the relationship between the detection sensitivity/recall and the pseudo string stability of a platoon under cyberattacks. We further find the the critical detection sensitivity under which one can maintain a pseudo stable string.

We adopt a variant of the well-known intelligent driver model (IDM), originally proposed by Treiber & Kesting (2012), namely the cooperative intelligent driver model (CIDM) from (Wang et al., 2020a) as our platooning model of choice, and implement our framework to compare the anomaly detection performance of the χ^2-detector and OCSVM in conjunction with EKF and AEKF. According to Treiber & Kesting (2012), IDM is suitable for describing the characteristics of automated driving, e.g. ACC. Although IDM has no explicit reaction time, it can also be easily extended to capture the communication delay, as described in the literature (Wang et al., 2020c,a). Note that compared with the CIDM model in (Wang et al., 2020a), in this chapter we further extend the IDM model to the setting of heterogeneous time delay. The CIDM with heterogeneous time delay can be described as follows,

$$\begin{aligned} \dot{v}_n(t) &= a^* \left(1 - \left(\frac{v_n(t)}{v_0} \right)^4 - \left(\frac{S^*(v_n(t), \bar{d}_n(t;\tau_1,\tau_2))}{\bar{g}_n(t;\tau_1,\tau_2)} \right) \right) \quad \text{with} \\ S^*(v_n(t), \bar{d}_n(t;\tau_1,\tau_2)) &= s_0 + T \cdot v_n(t) + \frac{v_n(t) \cdot \bar{d}_n(t;\tau_1,\tau_2)}{2\sqrt{a^* b^*}} \end{aligned} \tag{4.39}$$

where $\bar{d}_n(t;\tau_1,\tau_2)$ and $\bar{g}_n(t;\tau_1,\tau_2)$ are defined in (4.35), a^*, b^* represent the maximum acceleration and the maximum comfortable deceleration respectively, v_0 represents the desired free-flow velocity, $S^*(\cdot)$ is the desired clearance gap, s_0 denotes the minimum clearance gap in jammed traffic, and T is the desired time headway to follow the immediate leading vehicle.

v_0	33.33 m/s	Desired free-flow velocity
l	5 m	Vehicle length
T	1.1 s	Safety time headway
s_0	2	Minimum clearance gap
a^*	1 m/s^2	Maximum acceleration
b^*	2 m/s^2	Maximum comfortable deceleration

Table 4.2: CIDM parameters

4.4.1 Detection Performance under a Single Vehicle Attack

To measure the effectiveness of our proposed detection methodologies, we conduct sensitivity analysis over the Kalman filter configuration (i.e., with and without augmented state formulation), the anomaly detection methodology (i.e., the χ^2-detector and OCSVM), and time delays τ_1 and τ_2. We calculate the area under the curve (AUC) of each receiver operating characteristic (ROC) curve which summarizes the trade-off between the true positive rate and false positive rate ($1-$ specificity) for a predictive model at various threshold settings, by changing the values of γ of the χ^2-detector in (4.21), and parameter of OCSVM in (2.15). Since we are using an imbalanced dataset in which the number of non-anomalous cases is substantially higher than the number of anomalous cases, we further calculate the AUC score of the precision-recall curve, which summarizes the trade-off between the true positive rate and the positive predictive value (PPV, or precision) for a predictive model at various threshold settings, and is more suitable for imbalanced dataset.

Our experiments are based on the Safety Pilot dataset from the Safety Pilot Model Deployment (SPMD) program (Bezzina & Sayer, 2014) funded by the US department of Transportation, and collected in Michigan. The sampling frequency is 10 HZ (i.e. $\Delta t = 0.1$s). We sample the in-vehicle speed from the SPMD dataset with 4000 samples (400 seconds) for training set and 2000 samples (200 seconds) for testing set.

The testing scenario contains a vehicle platoon with 10 vehicles and each vehicle except for the platoon leader adopts the CIDM model in (4.39), and each vehicle except for the first two vehicles receive two cooperative leading vehicles' information. For simplicity, we set $\alpha_1 = \beta_1 = 0.8$ and $\alpha_2 = \beta_2 = 0.2$. The measurement vector z_n includes the location and velocity of the n-th vehicle. The platoon leader's trajectory is extracted from SPMD dataset, and the trajectory of the rest of the

platoon members are generated as the baseline based on the following rule:

$$\begin{aligned} x_n(k+1) = &\quad x_n(k) + v_n(k) \cdot \Delta t \\ v_n(k+1) = &\quad \Delta t \cdot f\left(v_n(k - \lfloor \tilde{\tau}_1/\Delta t \rfloor), \bar{g}_n(k; \lfloor \tilde{\tau}_1/\Delta t \rfloor), \lfloor \tilde{\tau}_2/\Delta t \rfloor)), \bar{d}_n(k; \lfloor \tilde{\tau}/\Delta t \rfloor), \lfloor \tilde{\tau}/\Delta t \rfloor)\right) \\ &\quad + v_n(k) + \epsilon_k \end{aligned} \tag{4.40}$$

where ϵ_k is sampled from a random variable to represent the inaccuracy caused by the flooring operation, and ϵ_k is sampled from a uniform distribution within range $[-0.1, 0.1]$. After obtaining the baseline data, we add Gaussian white noise with a variance of 0.3 to the baseline data to represent the measurement noise. Random anomalies are generated with 10% anomaly rate and injected into the trajectory data of the 5-th vehicle in the platoon using Algorithm 2.1 in Chapter 2. The anomaly magnitude for each type of anomaly is uniformly distributed within range $(0, 1]$, and the anomaly durations are also uniformly distributed from 1 to 20 time epochs.

Table 4.3: Detection performance of three models measuring on AUC scores of ROC curve and PR curve.

Time Delay	Scen 1: $\tau_1 = \tau_2 = 0$ s		Scen 2: $\mathbb{E}[\tilde{\tau}_1] = \mathbb{E}[\tilde{\tau}_2] = 0.5$ s		Scen 3: $\mathbb{E}[\tilde{\tau}_1] = \mathbb{E}[\tilde{\tau}_2] = 1.5$ s	
Metric	ROC AUC	PR AUC	ROC AUC	PR AUC	ROC AUC	PR AUC
χ^2 EKF	0.968 $\pm$ 0.018	0.922 $\pm$ 0.054	0.946 $\pm$ 0.018	0.895 $\pm$ 0.054	0.866 $\pm$ 0.016	0.820 $\pm$ 0.049
χ^2 ASEKF	0.968 $\pm$ 0.021	0.920 $\pm$ 0.056	0.953 $\pm$ 0.024	0.902 $\pm$ 0.060	0.938 $\pm$ 0.030	0.866 $\pm$ 0.068
OCSVM EKF	**0.977 $\pm$ 0.011**	**0.959 $\pm$ 0.020**	**0.974 $\pm$ 0.010**	**0.956 $\pm$ 0.019**	**0.964 $\pm$ 0.012**	**0.933 $\pm$ 0.019**
OCSVM ASEKF	0.970 $\pm$ 0.017	0.933 $\pm$ 0.019	0.966 $\pm$ 0.014	0.936 $\pm$ 0.026	0.959 $\pm$ 0.014	0.931 $\pm$ 0.024

The experiments are separately implemented into four models for ablation study, where model 1 is composed of a χ^2-detector in conjunction with EKF, model 2 is composed of a χ^2-detector in conjunction with ASEKF, model 3 is composed of OCSVM in conjunction with EKF, and model 4 is composed of OCSVM in conjunction with ASEKF. The CIDM parameters are set according to Table 4.2. Table 4.3 shows the performance of three models under three testing scenarios with different time delay settings, i.e. 0 seconds, 0.5 seconds, and 1.5 seconds, where $\tau_1 = \tau_2 = 0$ for scenario 1. For scenario 2 and scenario 3, we consider stochastic time delay with $\mathbb{E}[\tilde{\tau}_1] = \mathbb{E}[\tilde{\tau}_2] = 0.5$ and $\mathbb{E}[\tilde{\tau}_1] = \mathbb{E}[\tilde{\tau}_2] = 1.5$, respectively, with bounds of stochastic time delays $a_1 = a_2 = -0.1$, and $b_1 = b_2 = 0.1$. The measurement function $\mathcal{M}(\cdot)$ of ASEKF in equation (4.10) is defined as:

$$\mathcal{M}(\tilde{s}) = \begin{bmatrix} 1, 0, 1 \\ 0, 1, 0 \end{bmatrix} \cdot \tilde{s}$$

and the measurement function $\mathcal{M}(\cdot)$ of EKF is defined as:

$$\mathcal{M}(s) = \begin{bmatrix} 1, 0 \\ 0, 1 \end{bmatrix} \cdot s$$

The experiments indicate that the OCSVM with EKF fault detection method provides a significant improvement (up to 13.8% in PR AUC and 11.3% in ROC AUC) compared with the performance of the two χ^2-detector models, regardless of the value of time delay. Also, we observe that using augmented state formulation can significantly improve the performance of χ^2-detector under stochastic time delay (scenario 2 and scenario 3). However, when there is no time delay (scenario 1), using ASEKF does not lead to a better performance because there is no need to use the augmented state to compensate for potential bias caused by the time delay factors, and it introduces more uncertainties to the actual state which decreases the detection performance. Unlike the χ^2-detector, we observe a slight decrease of performance in OCSVM when it is combined with ASEKF. The reason is that OCSVM itself can learn the potential background bias in state-transition model and therefore the marginal benefit of using ASEKF is not prominent compared with the case for the χ^2-detector. Moreover, we observe that when using an augmented state formulation, the value of the augmented state is affected by the existence of anomalies and therefore the innovation distribution of the testing data is changed compared with that in the training data, which makes it more difficult for the trained OCSVM classifier to detect anomalies. Additionally, we observe a systematic deterioration of the detection performance as we increase the time delays, due to the fact that the time delays decrease the estimation accuracy of both EKF and ASEKF and therefore affect the detection performance.

4.4.2 Detection Performance under Multiple Vehicle Attacks

Next, we investigate the impact of cyberattacks on multiple vehicles in the platoon, and the effect of the detection and recovery. We assume each individual platoon member is equipped with a detector and is able to recover the true state only if it successfully detects an anomaly. All platoon vehicles are initialized at the steady state, i.e. with equilibrium clearance gap of g^* and equilibrium velocity of v^*, which can be obtained by having all vehicles in the platoon travel with the same constant velocity v^*,

$$g^* = v^*(s_0 + Tv^*)\left(1 - \left(\frac{v^*}{v_0}\right)^4\right)^{-0.5} - g_0 \tag{4.41}$$

where g_0 is the initial clearance gap between each pair of adjacent vehicles.

We consider a ten-vehicle platoon in a road-ring configuration (starting from ID 0 to ID 9). Each vehicle receives information from its immediate three leading vehicles, with $\alpha_1 = \beta_1 = 0.7$,

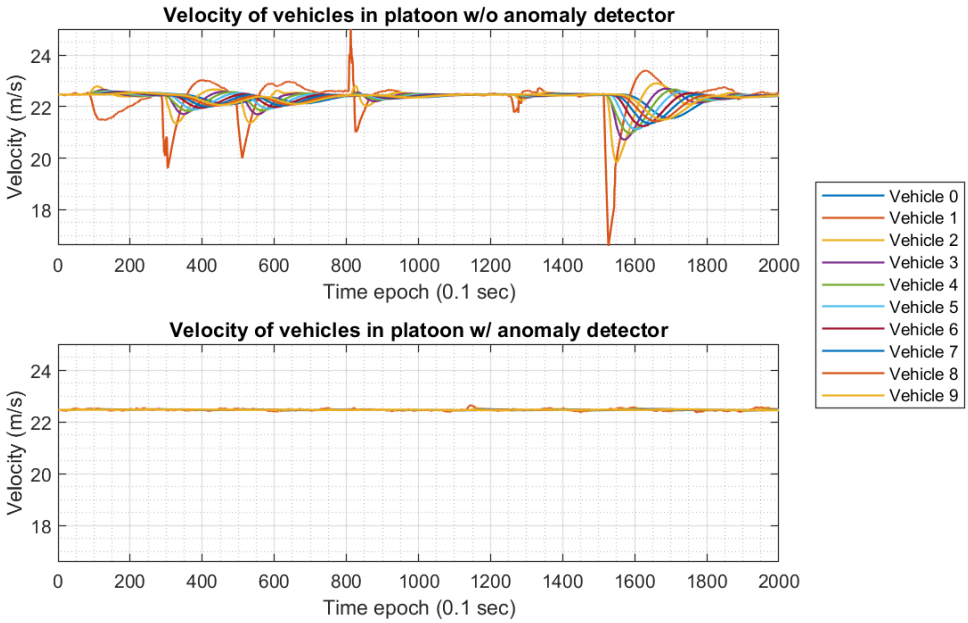


Figure 4.3: Vehicle velocity in platoon. Top: without detection and recovery. Bottom: with detection and recovery.

$\alpha_2 = \beta_2 = 0.2$, and $\alpha_3 = \beta_3 = 0.1$. Again, we use the same parameters in CIDM as presented in Table 4.2. The time delay is set to a fixed value of 0.5 seconds for both τ_1 and τ_2. A mixed set of anomaly types with anomaly rate of 10% were applied to five vehicles in the platoon, starting from the second vehicle to the sixth vehicle. The left and right subfigures in Figure 4.3 show the vehicle velocity with and without anomaly detection and recovery, respectively. We can observe a significant fluctuation of velocity under the attack scenario without detection and recovery, and conversely much smoother trajectories after detection and recovery.

Figure 4.4 shows the spacing error between each pair of adjacent vehicles in the platoon in the span of 200 seconds. The top subfigure shows the spacing error without anomaly detection, and the bottom subfigure shows the spacing error with anomaly detection and recovery. The spacing error $\Delta g_n(t)$ for the n-th vehicle at time t is defined as

$$\Delta g_n(t) = g_n(t) - g^* \tag{4.42}$$

and becomes zero when all vehicles in the platoon are in the equilibrium state. We can observe that when the platoon is under attack, there exist a lot of perturbations in terms of spacing error if no detector is deployed. Such perturbations are greatly reduced when using a detector, followed with a recovery step.

Figure 4.5 further shows the maximum spacing error with and without anomaly detection and

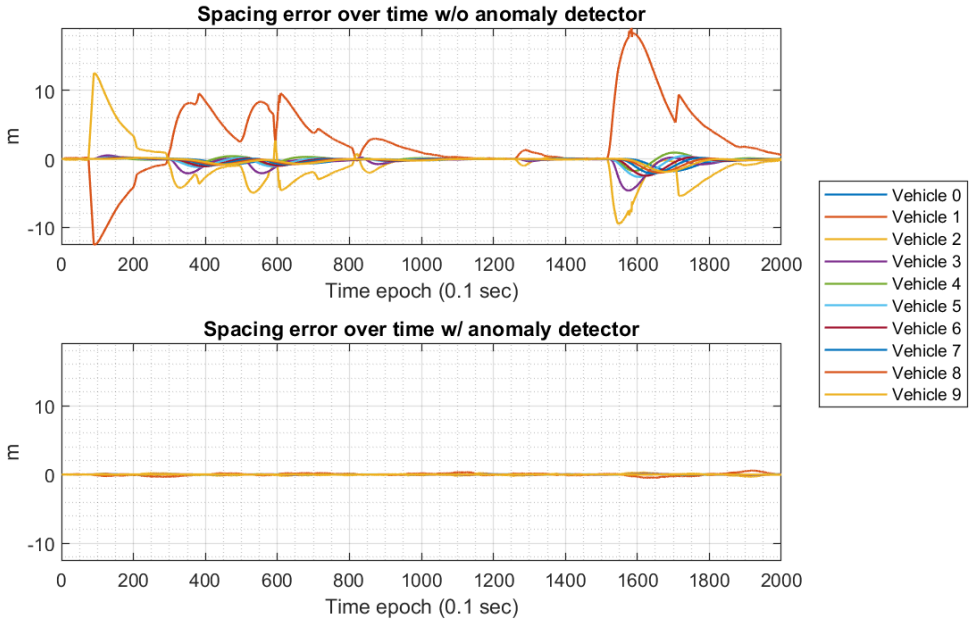


Figure 4.4: Spacing error over time under cyberattacks.Top: without anomaly detection and recovery. Bottom: with anomaly detection and recovery.

recovery. The maximum spacing error is defined as $\max_t |\Delta g_n(t)|$. According to Definition 4.1, the head-to-tail string stability can also be described in time domain:

$$\max_t |\Delta g_{N-1}(t)| < ... < \max_t |\Delta g_n(t)| < ... < \max_t |\Delta g_0(t)| \tag{4.43}$$

In Figure 4.5 we can observe an unstable vehicle string when there is no detector. However, the platoon stabilizes with detection and recovery.

4.4.3 Sensitivity Analysis on the Attack Parameters

We further conduct sensitivity analysis to study the effect of attack parameters on the platoon's string stability. Specifically, we analyze the effects of attack parameters $\tilde{A}, \tilde{B}, \tilde{C}, \tilde{\tau}_1$, and $\tilde{\tau}_2$ on the platooning model without anomaly detection:

$$\dot{v}_n(t) = f\left(v_n(t-\tilde{\tau}_1) + \tilde{A}, \bar{g}_n(t;\tilde{\tau}_1,\tilde{\tau}_2) + \tilde{B}, \bar{d}_n(t;\tilde{\tau}_1,\tilde{\tau}_2) + \tilde{C}\right) \tag{4.44}$$

In our simulations so far, the three attack parameters (noise terms) $\tilde{A}, \tilde{B}, \tilde{C}$ have been fixed. To further investigate the influence of attack intensity on platoon string stability, we experiment with multiple sets of attack parameters and keep the rest of parameters fixed, as in Table 4.2. We keep the same topology, where each vehicle will receive information from three predecessors, as shown

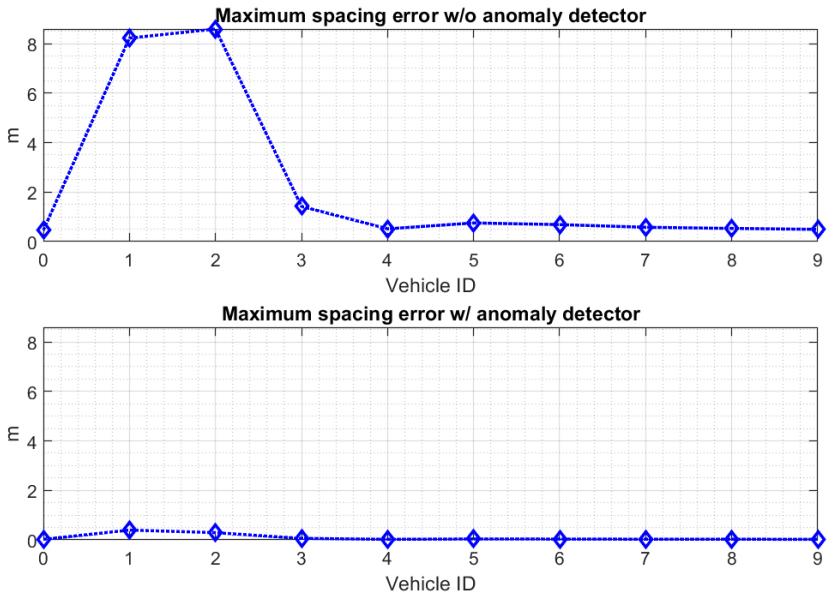


Figure 4.5: Maximum absolute spacing error under cyberattacks. Top: without anomaly detection. Bottom: with anomaly detection and recovery.

in Figure 4.2, and $\alpha_1 = \beta_1 = 0.7$, $\alpha_2 = \beta_2 = 0.2$, and $\alpha_3 = \beta_3 = 0.1$. The attack parameters tested for sensitivity analysis are listed in Table 4.4.

Parameter	Lower Bound	Upper Bound	Step Size
$\tilde{A}$	-15 m/s	15 m/s	0.2 m/s
$\tilde{B}$	-15 m	15 m	0.2 m
$\tilde{C}$	-15 m/s	15 m/s	0.2 m/s

Table 4.4: Sensitivity Analysis Parameters

Figure 4.6 shows the heat map of the largest eigenvalue of the transfer matrix given the combination of attack parameters $\tilde{A}$, $\tilde{B}$, and $\tilde{C}$. The color white indicates that the platoon will remain head-to-tail string stable for the given parameter combinations, while the colored region indicates loss of head-to-tail string stability. The color intensity indicates the magnitude of perturbation amplifications when the platoon is string unstable.

It can be observed that as $\tilde{A}$ increases from negative values to positive values, the unstable region shifts to the left. In an unstable platoon, when fixing the distance-targeted attack parameter $\tilde{B}$, to achieve the same level of peak amplification of perturbations in the platoon, the other two velocity-targeted attack parameters, $\tilde{A}$ and $\tilde{C}$, should be adjusted in opposite directions, i.e., increasing (decreasing) $\tilde{A}$, but decreasing (increasing) $\tilde{C}$. One explanation is that by increasing $\tilde{A}$, the ego

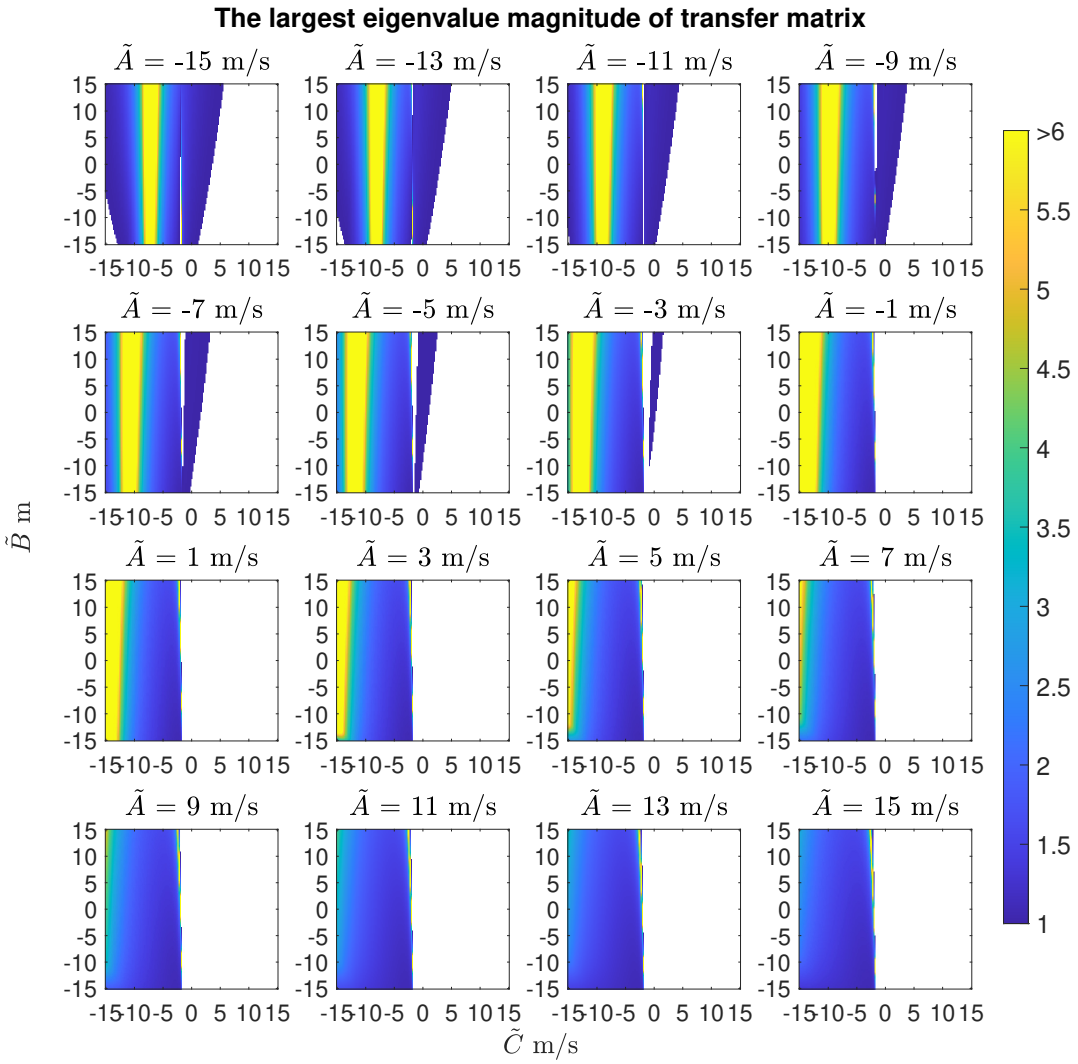


Figure 4.6: The white area in each subplot represents that given the set of attack parameters, the platoon can remain string stable. The color bar indicates the value of the largest eigenvalue of the transfer matrix.

vehicle falsely perceives its velocity higher than its actual velocity. Meanwhile if we do not change (or increase) $\tilde{C}$, which affects the relative velocity between the ego vehicle and its cooperative leaders, this indicates that the cooperative leaders are moving in higher velocities. Therefore within the current range of attack parameters, an attack that erroneously leads to a perceived higher velocity of leaders in the platoon compensates the damage done by an attack that erroneously leads to a perceived higher velocity of the ego vehicle, and could make the platoon more stable for some ranges of these attack parameters.

The relationship between the distance-targeted attack parameter and the two velocity-targeted attack parameters is even more complicated. However, we notice that when $\tilde{A} > -3$, there is

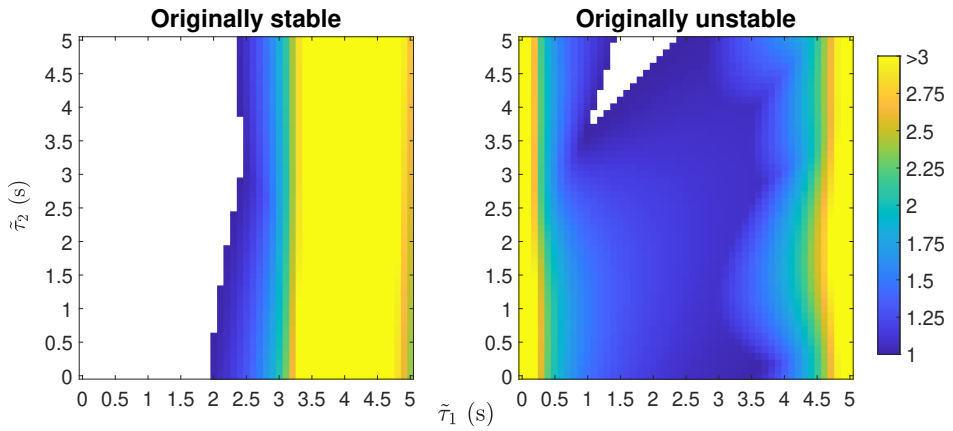


Figure 4.7: The left plot indicates the influence of the manipulated onboard time delay, $\tilde{\tau}_1$, and the manipulated communication time delay $\tilde{\tau}_2$ on platoon string stability when $\tilde{A} = \tilde{B} = \tilde{C} = 0$. The right plot shows the influence of $\tilde{\tau}_1$ and $\tilde{\tau}_2$ when the platoon is string unstable even in the absence of any attacks, and with $\tilde{A} = -3$ m/s, $\tilde{B} = -5$ m, $\tilde{C} = -11$ m/s. As a reference, the initial largest magnitude of eigenvalues of the transfer matrix when $\tilde{\tau}_1 = \tilde{\tau}_2 = 0$ is 5.2835. The maximum magnitude of eigenvalues with time delays is 5.3288. The white color represents the string stable region.

a trend that when fixing $\tilde{C}$, to make the platoon achieve the same level of peak amplification of perturbations requires $\tilde{A}$ and $\tilde{B}$ to change in the same direction, i.e., increasing (decreasing) $\tilde{A}$ and $\tilde{B}$ together. One possible explanation is that, under the current range of attack parameters, by increasing $\tilde{A}$, the ego vehicle perceives its velocity higher than its actual velocity. Increasing $\tilde{B}$, which affects the perceived clearance gap between the ego vehicle and its cooperative leaders, could illude the ego vehicle to actually drive faster, thereby making the platoon more unstable.

The influence of two time delay terms, $\tilde{\tau}_1$ and $\tilde{\tau}_2$, are also studied after fixing attack parameters $\tilde{A}$, $\tilde{B}$, and $\tilde{C}$. We use the same communication topology as shown in Figure 4.2 with $\alpha_1 = \beta_1 = 0.7$, $\alpha_2 = \beta_2 = 0.2$, and $\alpha_3 = \beta_3 = 0.1$. The time delays and attack parameters are set as follows: we first set $\tilde{\tau}_1 = 0$ and $\tilde{\tau}_2 = 0$, where no time delays exist in the system. For the first set of parameters, which create stable conditions, we set $\tilde{A} = \tilde{B} = \tilde{C} = 0$ such that the platoon becomes string stable without any time delay. In the second set of attack parameters, which provide unstable conditions, we set $\tilde{A} = -3$ m/s, $\tilde{B} = -5$ m, and $\tilde{C} = -11$ m/s to make the platoon string unstable even without any time delays. All combinations of $(\tilde{\tau}_1, \tilde{\tau}_2)$ are then tested in the range of $[0,\ 5]$ seconds with a step size of 0.1 seconds. The results are shown in Figure 4.7.

From Figure 4.7, we observe that when the dynamics model is attack-free, onboard time delay $\tilde{\tau}_1$ has more power in affecting platoon stability compared to the communication time delay $\tilde{\tau}_2$, as we can always find an unstable region by fixing τ_2 and adjusting $\tilde{\tau}_1$, but not vice versa. Furthermore, it appears that there is a threshold value $\tilde{\tau}_1^*$ ($\tilde{\tau}_1^* = 2$ seconds in this case) such that the platoon will remain string stable as long as $\tilde{\tau}_1 < \tilde{\tau}_1^*$, when $\tilde{\tau}_2$ is smaller than 5 seconds. An extreme case is

when $\tilde{\tau}_2 = \infty$, where the platoon is under a pure car-following model. In this case, $\tilde{\tau}_1$ represents the delay from the ego vehicle's onboard measurements, which in addition to the status of the ego vehicle provides information about its immediate leading vehicle. On the right subplot, we observe a more complex pattern. When fixing $\tilde{\tau}_2$, the peak amplification of perturbations can be altered greatly by changing $\tilde{\tau}_1$. However, except for the stable region in the upper part of figure, $\tilde{\tau}_2$ seems to have a more subtle impact on the peak amplification of perturbations. One interesting finding is that if the platoon is originally string unstable, increasing $\tilde{\tau}_1$ from 0 to 2.5 seconds can achieve a lower peak amplification of perturbations. This suggests that, when the platoon is unstable to begin with, the peak amplification of perturbations can be reduced by a slight increase of latency in onboard measurement. However, there may exist different trends for a broader range of attack parameters, because of the complex nature of the mutual impact of different attack parameters on the characteristics of stability region, which is characterized by the eigenvalues of the transfer matrix. Moreover, introducing larger time delays can eventually cause a crash, which is not reflected in stability analysis.

4.4.4 Pseudo String Stability Analysis

In this section, we conduct pseudo string stability analysis on model (4.35). One major contribution of this work is to bridge the gap between anomaly detection and platoon string stability. Since in practice the detection sensitivity/recall is not always 100%, as discussed in Section 4.3.6, the platoon model becomes a probabilistic model under detection uncertainties. Therefore, it is critical to find a minimum required detection sensitivity/recall such that any detector with a higher detection sensitivity can make the platoon maintain pseudo string stability. Here we present several case studies to find the desired detection rate that determines the pseudo string stability of the platoon.

First, in order to investigate the existence and characteristics of such critical detection sensitivity, we conduct sensitivity analysis on the pseudo string stability of a platoon with 10 vehicles under different detection sensitivities. We use the same communication topology as shown in Figure 4.2 with $\alpha_1 = \beta_1 = 0.7$, $\alpha_2 = \beta_2 = 0.2$, and $\alpha_3 = \beta_3 = 0.1$. According to inequality (4.38), to maintain pseudo string stability, one needs to make sure that the largest magnitude of eigenvalues of the transfer matrix is always smaller or equal to 1 across all frequency values. In this experiment, the parameters selected for CIDM remain the same as shown in Table 4.2. Time delays take values of $\tilde{\tau}_1 = 0$ and $\tilde{\tau}_2 = 0.5$. The attack parameters use $\tilde{A} = -5$ m/s, $\tilde{B} = 15$ m, and $\tilde{C} = -6$ m/s. From Figure 4.8, we can see that given the existing topology of three predecessors in a ten-vehicle platoon, as the anomaly detection sensitivity increases from 0 to 1, the platoon will incrementally reach pseudo string stability with the critical detection sensitivity $p^* = \sqrt[10]{\tilde{p}^*} \approx 0.985$.

In order to further investigate how attack parameters affect the critical detection sensitivity

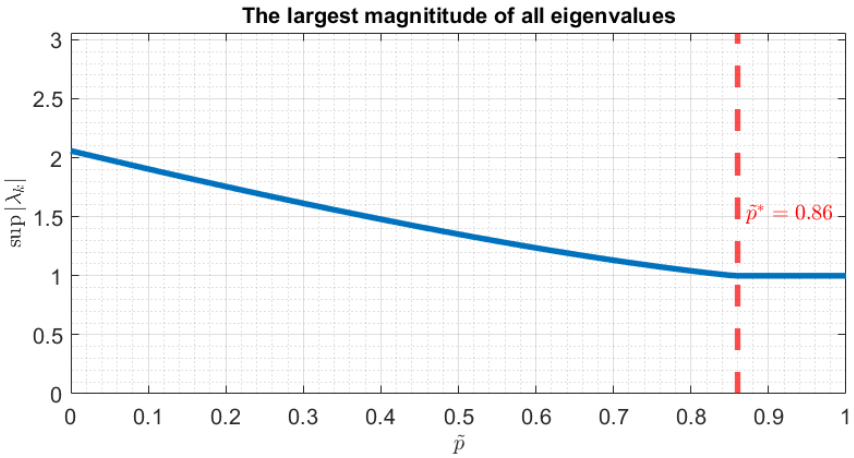
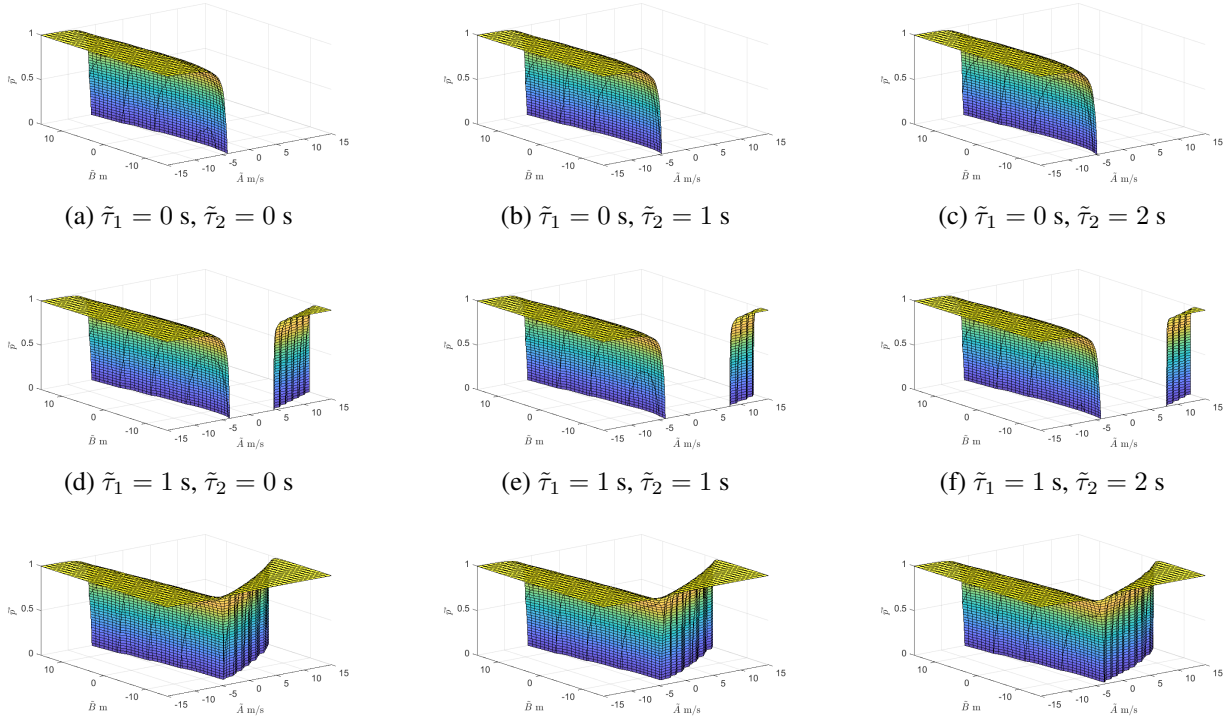


Figure 4.8: The blue curve represents the largest magnitude of all eigenvalues of the mean transfer matrix under detection uncertainties, i.e., $\tilde{p}$. The red dashed line indicates the critical point $\tilde{p}^* = (p^*)^{10} = 0.86$, when the largest magnitude becomes exactly 1, denoting a pseudo string stable platoon of 10 vehicles.



(a) $\tilde{\tau}_1 = 0$ s, $\tilde{\tau}_2 = 0$ s (b) $\tilde{\tau}_1 = 0$ s, $\tilde{\tau}_2 = 1$ s (c) $\tilde{\tau}_1 = 0$ s, $\tilde{\tau}_2 = 2$ s

(d) $\tilde{\tau}_1 = 1$ s, $\tilde{\tau}_2 = 0$ s (e) $\tilde{\tau}_1 = 1$ s, $\tilde{\tau}_2 = 1$ s (f) $\tilde{\tau}_1 = 1$ s, $\tilde{\tau}_2 = 2$ s

(g) $\tilde{\tau}_1 = 2$ s, $\tilde{\tau}_2 = 0$ s (h) $\tilde{\tau}_1 = 2$ s, $\tilde{\tau}_2 = 1$ s (i) $\tilde{\tau}_1 = 2$ s, $\tilde{\tau}_2 = 2$ s

Figure 4.9: Critical detection sensitivity $\tilde{p}^*$ under different attack parameters $\tilde{A} \in [-15, 15]$ m/s and $\tilde{B} \in [-15, 15]$ m by fixing $\tilde{C} = -1$ m/s. Each subfigure contains a hyperplane which represents the critical detection sensitivity $\tilde{p}^*$ under different time delay factors $\tilde{\tau}_1$ and $\tilde{\tau}_2$ in range $\{0, 1, 2\}$ seconds.

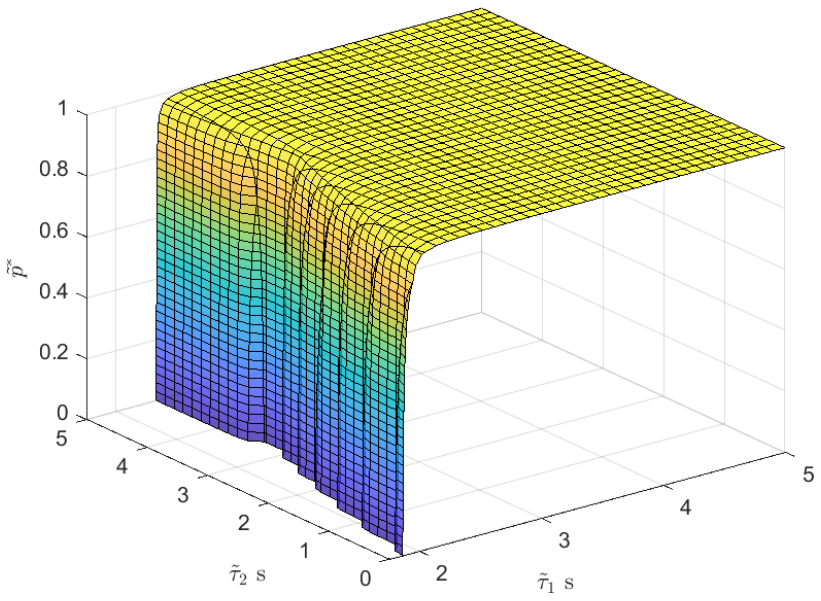


Figure 4.10: Critical detection sensitivity $\tilde{p}^*$ under different attack parameters $\tilde{\tau}_1$ and $\tilde{\tau}_2$ by fixing the values of $\tilde{A}$, $\tilde{B}$, and $\tilde{C}$ to be 0.

value, we conduct sensitivity analysis by varying the different attack parameters, including $\tilde{A}$, $\tilde{B}$, $\tilde{\tau}_1$, and $\tilde{\tau}_2$, and calculating the corresponding critical $\tilde{p}^*$. Specifically, we consider a platoon with 10 vehicles and with 3 predecessors, where $\alpha_1 = \beta_1 = 0.7$, $\alpha_2 = \beta_2 = 0.2$, and $\alpha_3 = \beta_3 = 0.1$, following the CIDM parameters in Table 4.2. Figure 4.9 shows the critical detection sensitivity $\tilde{p}^*$ that maintains pseudo string stability of the platoon. In each subfigure, the hyperplane represents $\tilde{p}^*$ in z-axis for a range of $\tilde{A} \in [-15, 15]$ in x-axis and $\tilde{B} \in [-15, 15]$ in y-axis by fixing $\tilde{C} = -1$. We generate 9 subfigures by considering all combinations of $\tilde{\tau}_1 \in \{0, 1, 2\}$ seconds and $\tilde{\tau}_2 \in \{0, 1, 2\}$ seconds. We first observe that in the given range of parameters, both attack parameters $\tilde{A}$ and $\tilde{B}$ determine the value of $\tilde{p}^*$, whereas the effect of destabilization is more prominent for the attack parameter $\tilde{A}$ when $\tilde{\tau}_1 = 0$. From subfigures 4.9(d)-4.9(i), when $\tilde{A} > -3$ m/s, we observe the same trend as in Figure 4.6, i.e., when $\tilde{A} > -3$ m/s, in order to maintain the same level of peak amplification of perturbations, one needs to increase (decrease) $\tilde{A}$ and $\tilde{B}$ together. Each column of the subfigures in Figure 4.9 indicates that as we increase the value of $\tilde{\tau}_1$, we obtain a larger pseudo unstable region where we have non-zero critical detection sensitivity $\tilde{p}^*$. The destabilization effect of $\tilde{\tau}_2$ is less prominent than $\tilde{\tau}_1$, as we only observe a minimal change of hyperplane distribution in each row of Figure 4.9, which works in concert with our observation in Figure 4.7.

In Figure 4.10, we investigate the standalone impact of manipulated delay factors $\tilde{\tau}_1$ and $\tilde{\tau}_2$ on the critical detection sensitivity. Figure 4.10 represents the critical detection sensitivity $\tilde{p}^*$ (z-axis) that maintains the pseudo string stability under different time delay factors $\tilde{\tau}_1 \in [0, 5]$ seconds (x-axis) and $\tilde{\tau}_2 \in [0, 5]$ seconds (y-axis), where we fix the values of $\tilde{A}$, $\tilde{B}$, and $\tilde{C}$ to 0. Note that the

projection of the hyperplane on the x-y plane is equivalent to the left figure in Figure 4.7 when $\tilde{\tau}_1$ and $\tilde{\tau}_2$ are within $[0, 3]$ seconds, which indicates the pseudo unstable region of the platoon when we do not use any anomaly detector.

4.5 Conclusion

CAVs can receive and utilize information from multiple sources to form vehicular platoons. However, literature has demonstrated that a CAV is more vulnerable to cyberattacks, as it has more attack surfaces. Existing literature either only investigates the impact of cyberattacks on platoons or defense methodologies, such as detection and protocol design. Instead, in this chapter, we develop a comprehensive framework to model the impact of cyberattacks on platoons and to detect sensor measurement anomalies caused by either malicious attacks or sensor faults. Specifically, we propose a general platoon dynamics model under heterogeneous time delays, and design a state-space model for filtering and anomaly detection by utilizing cooperative vehicles' information. We further extend this model to consider stochastic time delays and show its impact on the state-space model. In order to investigate the impact of cyberattacks on platoons, we first conduct string stability analysis of the proposed platoon dynamics model. To the best of our knowledge, this is the first head-to-tail string stability analysis under heterogeneous time delay. Next, we propose a new definition for string stability under cyberattacks and model uncertainties, which we call pseudo string stability.

For vehicle state estimation, we show that under stochastic time delay, there may exist potential bias in the process noise of our proposed state-space model. To compensate for this, we propose an augmented state extended Kalman filter (ASEKF) for vehicle state estimation. For anomaly detection in the vehicle sensor measurements, we adopt two anomaly detectors, namely the χ^2-detector and the one class support vector machine (OCSVM), in conjunction with ASEKF. We conduct extensive experiments to demonstrate the effectiveness of our proposed detection framework. Specifically, we conduct an ablation study showing that an extended Kalman filter with an OCSVM detector achieves the best performance, whereas an augmented state formulation can significantly boost the performance of the χ^2-detector under time delay. Our experiments also show a negative impact of the time delay on the overall anomaly detection performance.

To study the impact of cyberattacks on the platoon's string stability, we conduct sensitivity analysis on the attack parameters. We observe certain relationships between the distance- and velocity-targeted attack parameters in affecting the peak amplification of perturbations in the platoon. In our experiments, we further investigate the pseudo string stability of platoons under different detection sensitivities and obtain the critical detection sensitivity to ensure a pseudo stable vehicle string.

The chapter is subject to certain limitations. In our experiments, similar to previous studies in

the literature, due to the paucity of real-world anomalous CAV data, the anomalous instances in the sensor data are simulated with a mix of five types of anomalies. This implicitly imposes an assumption on the characteristics of the anomalous data. To partially address this limitation, we adopt a OCSVM model to learn the detection threshold by merely learning from normal data. It may be beneficial to test for novel anomaly types using real-world anomalous data to more accurately measure the effectiveness of our proposed detection methods.

CHAPTER 5

Adversarial Online Learning with Variable Plays in the Pursuit-Evasion Game: Theoretical Foundations and Application in Connected and Automated Vehicle Cybersecurity

5.1 Introduction

Currently, the world is experiencing an evolution from the traditional transportation system to the next generation of intelligent transportation systems (ITS). ITS aims to satisfy the ever-increasing need for mobility in major cities, which has caused growing traffic congestion, air pollution, poor user experience and crashes. Developing a sustainable intelligent transportation system requires better usage of the existing infrastructures and their seamless integration with information and communication technologies (ICT). Enabled by the recent findings in the areas of telecommunications, electronics, and computing capabilities in recent decades, the subsystems (infrastructures and vehicles) in ITS are expected to interoperate and communicate with each other, in order to provide a better and safer traveling experience (Ibanez et al., 2015).

The interconnection between the infrastructures and the vehicles relies on various types of sensors to provide state information and situational awareness. However, this has also increased the vulnerability of these advanced systems to cyber attacks. For instance, recently there have been demonstrated cyber attacks on vehicle sensors in (Cao et al., 2019a,b), where the authors used optimization-based approaches to fool the light detection and ranging (LiDAR) sensors on the vehicle. At the system level, the infrastructures and the vehicles can be viewed as individual nodes in a large interconnected network, where a single malicious attack on a subset of sensors of one node can easily propagate through this network, affecting other network components (e.g., other vehicles, traffic control devices, etc.). For instance, Feng et al. (2018) demonstrated that by sending falsified data to actuated and adaptive signal control systems, a malicious hacker could increase the total system delay in a real-world corridor. Therefore, there is an increasing need for cyber security solutions, especially for sensor security solutions, to enhance the safety and reliability of the entire system.

Cyber security is an extremely broad topic. However, previous work on cyber security in the realm of ITS mainly focuses on either attack or the defense strategies. For instance, there exists a large body of research illustrating the potential risks of connected and automated vehicle (CAV) technologies that result in anomalous/false information (Petit & Shladover, 2014; Checkoway et al., 2011; Weimerskirch & Gaynier, 2015; Yan et al., 2016). In the case of CAV sensor security, several critical sensors are illustrated in (Parkinson et al., 2017), including differential global positioning systems (GPS), inertial measurement units, engine control sensors, tyre-pressure monitoring systems (TPMS), LiDAR, and camera. Meanwhile, CAVs require more engine control units (ECUs) and many features of CAVs require complex interactions between multiple ECUs, which may potentially expose more vulnerabilities compared to non-CAVs. There also exist several studies assessing the potential threats on the transportation infrastructure (Feng et al., 2018; Fok, 2013; Kelarestaghi et al., 2018). For example, field devices such as traffic signals and roadside units are susceptible to tampering. The aforementioned literature illustrates the potential threats of sensor attacks to connected transportation systems.

Besides threat detection, prevention is normally recognized as one of the best defense strategies against malicious hackers or attackers. In order to deploy better prevention mechanisms, behaviors of both the attacker and the defender have to be considered so that the attack profile can be predicted. There is a gap in the literature in considering both the attacker and the defender and the adaptive interactions between them when devising defense strategies, which this chapter aims to bridge.

Moreover, as more sensors are mounted aboard CAVs or installed on the transportation infrastructure, it becomes more difficult to monitor the sensors continuously, mainly due to limited resources. Although there is a large body of literature addressing sensor security in ITS (Van Wyk et al., 2019; Marchetti et al., 2016; Müter & Asaj, 2011) including the first three chapters of the thesis, most of them mainly focus on sensor intrusion/anomaly detection without attack profile analysis, which considers which sensor is more vulnerable and should be protected. In this chapter, we address this by modeling attacker and defender behaviors in a game theoretical framework. Specifically, instead of considering intrusion/anomaly detection for all sensors in the system, we model attack and defense behaviors in order to predict which subset of sensors are more likely to be compromised. To be more practical, we consider a dynamic resource constraint for the defender. We model this problem as a sequential evasion-and-pursuit game between two players. Consider the intrusion monitoring system of a sensor network as the defender. At each time, the defender selects a subset of sensors to scan, while the number of selected sensors changes based on the environment and scanning history, among other factors. Meanwhile, a hacker, considered as the attacker, attempts to select a sensor to compromise without being scanned by the defender. We assume that both the attacker and the defender are able to learn their opponent's behavior adaptively and with only partial information over time, and investigate the the resulting decision problem.

The main contributions of this work are as follows: First, in order to predict the attack profile, we model the behaviors of the attacker and the defender as the adversarial (or non-stochastic) multi-armed bandit (MAB) problem and the multi-armed bandit problem with variable plays (MAB-VP), where the two players are playing a constant-sum game against each other. To the best of our knowledge, this is the first study of MAB-VP in the non-stochastic setting. Second, we derive conditions under which a Nash equilibrium of the strategic game exists. For the defender, we provide an exponential-weighted algorithm, which is shown to have sublinear pseudo-regret. Finally, we consider a more realistic setting where the rewards are heterogeneous among different sensors, and derive lower and upper bounds on the attacker's average reward.

5.2 Literature Review

In this chapter, we explore online learning algorithms in the class of adversarial or non-stochastic multi-armed bandit (MAB) problems. The adversarial MAB problem was first addressed by Auer et al. (1995), where they also proposed the well-known exponential-weight algorithm for exploration and exploitation (Exp3). Exp3 runs the Hedge algorithm, which was originally proposed by Freund & Schapire (1997) as a subroutine. Since then, there have been several extensions to this class including the online shortest path problem (György et al., 2007), routing games (Nisan et al., 2007), bandit online linear optimization (Abernethy et al., 2008), and combinatorial bandits (Cesa-Bianchi & Lugosi, 2012).

The multi-play multi-armed bandit (MPMAB) problem is another research direction for MAB. In this extension, a fixed number of resources (i.e., arms) are allocated at each time step. The MPMAB has attracted a lot of interest and several studies have been conducted along this direction (Anantharam et al., 1987; Agrawal et al., 1990; Komiyama et al., 2015; Xia et al., 2016). However, most of these studies only focus on a stochastic setting. There is much less concentration on the adversarial MPMAB problem: Cesa-Bianchi & Lugosi (2012) considered combinatorial bandits in the adversarial setting, where they proposed the ComBand algorithm. This algorithm has a sublinear regret in $O\left(M^{\frac{3}{2}}N\sqrt{TN\ln N}\right)$, with time and space complexities of $O\left(MN^3\right)$ and $O\left(K^3\right)$, respectively, where M is the number of resources (or arms selected) at each time, T is the number of iterations, and N is the number of possible actions. Following this work, Uchiya et al. (2010) proposed an extension of Exp3, Exp3.M, which runs in $O\left(N(\log M+1)\right)$ time and $O(N)$ space, and suffers at most $O\left(\sqrt{MTN\log\left(N/M\right)}\right)$ regret. However, the aforementioned algorithms only consider a fixed number of arms to be played at each time.

Another branch of literature closely related to this chapter is on the topic of discrete search. For instance, Song & Teneketzis (2004) considered the discrete search problem with multiple sensors, and derived the optimal search strategies that maximize the total probability of successful search in

a finite horizon. They further discussed the relationship to MPMAB, where it can be viewed as a finite-horizon deterministic MPMAB with a discount factor equal to one. However, they did not consider the adversarial setting and variable plays.

Only a limited number of studies have considered variable plays. Fouché et al. (2019) proposed a scaling algorithm combined with a MAB algorithm , which they call the S-MAB algorithm. In this algorithm, the number of arms played at each time changes in order to satisfy an efficiency constraint. However, although the authors considered a dynamic environment, the S-MAB algorithm uses a stochastic setting, where they assume an unknown distribution of reward for each arm. Another work addressing the variable plays problem was done by Lesage-Landry & Taylor (2017), where they extended the stochastic MAB to stochastic plays setting, i.e. the number of arms to play evolves as a stationary process. Both these studies only considered a stochastic setting, and did not conduct any game strategy analysis.

Although there is a wealth of research on using game theory in the transportation literature, very few studies applied game theory in ITS cybersecurity. Sedjelmaci et al. (2019) conducted a survey on recent studies utilizing game theory to protect ITS from attacks, which is to the best of our knowledge the only survey paper on this topic. However, without considering the adaptive behavior of opponents, the current literature mostly models the cybersecurity problem as a non-repeated game, such as the Stackelberg security games (SSG) (Kiekintveld et al., 2009; Sinha et al., 2015), zero-sum games (Alpcan & Buchegger, 2010; Mejri et al., 2016), or Bayesian games (Bahamou et al., 2016; Sedjelmaci et al., 2016). The solutions from these types of models are typically in the form of equilibria with an implied assumption that the players have knowledge of their opponent's actions/beliefs. Instead, we formulate this cybersecurity problem as a sequential pursuit-evasion game, which is also in the realm of algorithmic learning theory. There have been several studies of the pursuit-evasion problem (Vidal et al., 2002; Navda et al., 2007; Wang & Liu, 2015). However, they either lack robustness against adaptive changes in the adversarial behavior, or do not consider multiple plays, variable plays, dynamic resource allocation, or heterogeneous rewards.

Since the behavior of the adversarial opponent usually cannot be described in a stochastic way, in this chapter we study the MAB-VP problem in a non-stochastic setting, where we propose the Exp3.M with variable plays (Exp3.M-VP) algorithm. Next, we consider a game setting for two players, and show that a Nash equilibrium of the strategic game exists. Finally, we consider heterogeneous rewards for both players and derive lower and upper bounds for the attacker's average reward. Numerical analyses are conducted in order to further demonstrate our results.

Table 5.1: Table of Notation

$\alpha_k(t)/\beta_k(t)$	$\triangleq$	marginal probability that the attacker compromises/the defender scans location k at time t
$x_k(t)/y_k(t)$	$\triangleq$	indicator variable of whether the defender/attacker selects the location k at time t
I_t/J_t	$\triangleq$	index of the locations where the attacker compromises/the defender scans at time t
M_t	$\triangleq$	number of locations scanned by the defender at time t
a/b	$\triangleq$	lower/upper bound of M_t
$r(t)/s(t)$	$\triangleq$	single step reward of the attacker/defender
$\omega(t)/\theta(t)$	$\triangleq$	private randomization device of the attacker/defender
π_t/γ_t	$\triangleq$	control policy of the attacker/defender
T	$\triangleq$	finite time horizon
N	$\triangleq$	total number of locations
$\mathcal{N}$	$\triangleq$	index set of N locations
$\mathcal{C}$	$\triangleq$	index set of arbitrary locations

5.3 System Model and Problem Formulation

5.3.1 System Model

Consider the repeated pursuit-evasion game between an attacker and a defender in discrete time. At each time step t, the attacker selects one of the N locations, indexed by the set $\mathcal{N} = \{1, 2, ..., N\}$, to hide in (e.g., compromise a sensor), while the defender searches M_t locations simultaneously, where $1 \leq a \leq M_t \leq b < N$. The behaviors of the attacker and the defender are described by their respective set of marginal probabilities $\alpha(t) = (\alpha_k(t))_{k\in\mathcal{N}}$ and $\beta(t) = (\beta_k(t))_{k\in\mathcal{N}}$, where $\alpha_k(t)$ and $\beta_k(t)$ are the respective probabilities that the k-th location is chosen by the attacker and the defender at time t. Note that $\alpha(t)$ and $\beta(t)$ represent the adversarial behavior with respect to one's opponent at time t, where they can describe randomized strategies of the players, or a probabilistic belief held by one side about the likelihood of an action by the other side.

Define two sets of binary variables $x_k(t)$ and $y_k(t)$ such that $x_k(t) = 1$ if the defender does not search location k at time t, and $x_k(t) = 0$ otherwise. Similarly, $y_k(t) = 1$ if the attacker compromises the location k at time t, and $y_k(t) = 0$ otherwise. When the attacker (defender) does not know the type of algorithm/strategy the opponent uses, it may regard the $x_k(t)$ ($y_k(t)$) as a predetermined but unknown number. When the attacker (defender) does have this information, it may regard the $x_k(t)$ ($y_k(t)$) as a random variable, where $P(x_k(t) = 0) = \beta_k(t)$ (resp. $P(y_k(t) = 1) =$

$\alpha_k(t)$). The game is played in a sequence of trials $t = 1, 2, ..., T$. In this work we consider the case that neither the attacker nor the defender knows the strategy adopted by the other player. As will be discussed later, they have to choose the location based on the their historical rewards.

5.3.2 Problem Formulation: Partial Information Game

In this chapter we consider the scenario where both players have limited information on the adaptive behavior of their opponent. Define $\pi = (\pi_t, t = 1, 2, ...)$ as the control policy of the attacker, and let Π denote the policy space. Denote the location selection (action) sequence as $I = (I_t, t = 1, 2, ...)$ under policy π and $|I_t| = 1$. At each time and under policy π_t, the attacker chooses one location $I_t \in \mathcal{N}$ to attack, i.e.,

$$I_t = \pi_t \left(x_I^{[t-1]}, I^{[t-1]}, \omega(t) \right) \tag{5.1}$$

where $x_I^{[t-1]} := (x_{I_1}(1), ..., x_{I_{t-1}}(t-1))$, and $I^{[t-1]}$ is similarly defined. $(\omega(t), t = 1, 2, ...)$ denotes the randomized strategy of the attacker. Let $x_k(t)$ be the state of location k for the attacker at time t. Then the attacker scores the corresponding reward $r^I(t) = x_{I_t}(t)$. The attacker observes only the reward $r^I(t)$ for the chosen action I_t.

The attacker receives an expected reward $\mathbb{E}[r^I(t)] = 1 - \beta_{I_t}(t)$ at time t, which is the mean number of successful attacks at the chosen location. Note that in this section we consider a homogeneous reward across all locations; however, heterogeneous location-dependent rewards are considered in Section 5.7. In this chapter, we assume a 100% success rate for both attacks and detection attempts. Then, within the time window $\{t, t = 1, 2, ..., T\}$, the attacker considers the following maximization problem,

$$\underset{\pi \in \Pi, I_t \in \mathcal{N}}{\text{maximize}} \ \mathbb{E} \left\{ \sum_{t=1}^{T} x_{I_t}(t) \right\} \tag{5.2}$$

where the expectation is with respect to the randomness of the system state and the mixed-strategy of the attacker.

We assume that the defender can scan M_t locations at time t. Define $\gamma = (\gamma_t, t = 1, 2, ...)$ as the control policy of the defender, and let Γ denote the defender's policy space. Denote the location selection (action) sequence as $J = (J_t, t = 1, 2, ...)$ under policy γ. At each time and under policy γ_t, the defender scans M_t locations, denoted as set $J_t \subset \mathcal{N}$ and $|J_t| = M_t$, based on their historical search and rewards, i.e.,

$$J_t = \gamma_t \left(y_J^{[t-1]}, J^{[t-1]}, M_t, \theta(t) \right) \tag{5.3}$$

where $y_J^{[t-1]} := (y_{J_1}(1), ..., y_{J_{t-1}}(t-1))$ with $J^{[t-1]}$ similarly defined, and $(\theta(t), t = 1, 2, ...)$ denotes the randomized strategy of the defender. Let $y_k(t)$ be the state of location k for the

defender at time t. The defender also observes only the rewards $\sum_{j\in J_t} y_j(t)$ of the selected action J_t. Denote the total rewards at time t of the defender given the location selection sequence J as $s^J(t) = \sum_{j\in J_t} y_j(t)$. The defender therefore receives the expected reward $\mathbb{E}[s^J(t)] = \sum_{j\in J_t} \alpha_j(t)$ at time t. This expected reward represents the mean number of detected attacks among M_t number of scanned locations.

We assume that the number of arms M_t the defender plays at each time is determined by a scaling function, i.e. $f : \mathbb{R}^{N+1} \to \{a, a+1, ..., b\}$, of the d-moving average of the rewards of each arm, where a and b are integers, and $1 \leq a \leq b < N$. We also assume that M_t is a function of the environment constraint L_t, since in reality checking a location (e.g., scanning a specific sensor/unit in a CAV) may consume resources. Then, given the time horizon T, the defender is trying to solve the following constrained optimization problem:

$$\underset{\gamma\in\Gamma, J_t\subset\mathcal{N}}{\text{maximize}}\ \mathbb{E}\left\{\sum_{t=1}^{T}\sum_{j\in J_t} y_j(t)\right\} \tag{5.4a}$$

$$\text{s.t. } M_t = f(\hat{y}^d(t), L_t) \tag{5.4b}$$

$$|J_t| = M_t \tag{5.4c}$$

where $\hat{y}^d(t) := (\hat{y}_1^d(t), \hat{y}_2^d(t), ..., \hat{y}_N^d(t))$, and $\hat{y}_i^d$ is the d-moving average of the rewards of each arm i. Using a moving average of reward can allow us to capture the historical reward while at the mean time capturing the dynamic change of the reward for each location, allowing the scaling function to adjust the number of arms to play each time. The expectation is with respect to the randomness of the system state and the mixed strategy of the defender. Note that there is no requirement for the scaling function f, other than it needs to be bounded by integers a and b. Furthermore, L_t can be an arbitrary integer between a and b, thereby capturing any set of environmental conditions.

When the defender knows the type of strategy the attacker uses, it may regard $y_j^J(t)$ as stochastic, i.e. assuming the attacker chooses location j with probability $P(y_j^J = 1) = \alpha_j(t)$. Note that this is different from the stochastic MAB setting where a fixed (time-invariant) distribution of rewards for each arm is assumed. However, here we do not assume neither the defender nor the attacker have information about their opponent's strategy. Hence, the difficulty is that the defender can only estimate $\alpha_j(t)$ by imposing an arbitrary belief on the adversarial behavior based on previous observations and rewards. Furthermore, here, we do not make any assumptions about the distribution of $\alpha_j(t)$.

Algorithm 5.1: **Exp3.M-VP**

Parameter: $\eta \in (0, 1]$
Initialization: $w_i(1) = 1$ for $i = 1, 2, ..., N$
for $t = 1, 2, ..., T$ **do**
  Receive the number of arms to play at each round M_t.
  if $\max_{j \in \mathcal{N}} w_j(t) \geq \left(\frac{1}{M_t} - \frac{\eta}{N}\right) \sum_{i=1}^{N} w_i(t)/(1-\eta)$ **then**
    Decide κ_t such that

$$\frac{\kappa_t}{\sum_{w_i(t) \geq \kappa_t} \kappa_t + \sum_{w_i(t) < \kappa_t} w_i(t)} = \left(\frac{1}{M_t} - \frac{\eta}{N}\right) / (1 - \eta).$$

    Set $S_0(t) = \{i : w_i(t) \geq \kappa_t\}$.
    Set $w'_i(t) = \kappa_t, \forall i \in S_0(t)$.
  else
    Set $S_0(t) = \emptyset$.
  end if
  Set $w'_i(t) = w_i(t), \forall i \in S_0^c(t)$.
  Set $\hat{\alpha}_i(t) = M_t \left((1-\eta)\frac{w'_i(t)}{\sum_{j=1}^{N} w'_J(t)} + \frac{\eta}{N}\right)$.
  Set $J_t =$ **DepRound**$(M_t, (\hat{\alpha}_1, \hat{\alpha}_2, ..., \hat{\alpha}_N))$.
  Observe rewards $y_i(t) \in [0, 1]$ for $i \in J_t$.
  for $i = 1, 2, ..., N$ **do**

$$\hat{y}_i(t) = \begin{cases} y_i(t)/\hat{\alpha}_i(t) & \text{if } i \in J_t, \\ 0 & \text{otherwise.} \end{cases}$$

$$w_i(t+1) = \begin{cases} w_i(t) \exp(M_t \, \eta \, \hat{y}_i(t)/N) & \text{if } i \in S_0^c(t), \\ w_i(t) & \text{otherwise.} \end{cases}$$

  end for
end for

5.4 Algorithms for the Attacker and the Defender

We assume the attacker adopts Exp3 proposed by Auer et al. (Auer et al., 1995). (However, as we are going to show later in Section 5.6, the equilibrium of the two-player game does not depend on any proprieties of the algorithm other than a no-regret guarantee.) The Exp3 algorithm uses an efficient and randomized policy to select only one arm at each time t. The adversarial single play bandit problem is closely related to the problem of learning to play an unknown repeated matrix game. In this setting, a player without prior knowledge of the game matrix is to play the game repeatedly against an adversary with complete knowledge of the game and unbounded computational power. The basic idea of Exp3 is that at each time the player uses a randomized policy such that the adversarial player cannot know the exact choice of the player before she/he plays. For the details of Exp3, refer to the Appendix A.1.

Unlike the attacker who selects a single location to attack, we assume the defender can search multiple number of locations, which may vary at each time. Both sides seek to maximize their respective total rewards. At the beginning of a time step, each side needs to decide which location(s) to target, and cannot change their selection until the next time step. We develop a variable-play extension of the Exp3.M algorithm for the defender, which we call Exp3.M-VP, as detailed in Algorithm 5.1. In the Exp3.M-VP algorithm, let S denote the set of selected locations, and let S^c define its complement set. Under the non-stochastic assumption and at each time step, the Exp3.M-VP algorithm consists of the following two procedures:

1. Receive M_t, which is determined by the scaling function f and could be based on the environment constraint L_t as well as the historical rewards $\hat{y}^d(t)$ at time t, among other factors. Note that function f can take any form, and defining its exact form is outside the scope of this chapter. Here, we assume M_t is provided.
2. Apply an adversarial MPMAB algorithm which selects M_t arms (locations) to play.

For the second procedure, we use the Exp3.M algorithm as a subroutine of the Exp3.M-VP algorithm. The Exp3.M is proposed by Uchiya et al. (Uchiya et al., 2010) and is an extension of the algorithm Exp3 for the adversarial MPMAB setting. In contrast to the Exp3 algorithm which selects one arm at each time, Exp3.M randomly selects a fixed number of M arms at each time. Note that both Exp3 and Exp3.M suffer from sublinear (weak) regret, or no-regret. In order to make sure that the probability of selecting location i by DepRound at step 12, i.e. $\hat{\alpha}_i(t)$, does not exceed 1, the Exp3.M-VP algorithm checks whether all $w_j(t)$'s are less than $\left(\frac{1}{M_t} - \frac{\eta}{N}\right) \sum_{i=1}^{N} \frac{w_i(t)}{(1-\eta)}$ at step 5. If that is the case, $\hat{\alpha}_i(t)$ calculated at step 11 will be less than 1 for all $i = 1, 2, ..., N$ without any weight modification, and the set $S_0(t)$ is set to $\emptyset$ at step 8. Otherwise, all the actions i with $w_i(t) \geq \kappa_t$ are classified into $S_0(t)$ and set to κ_t at step 6. Doing this, we have $\hat{\alpha}_i(t) = 1$ for all $i \in S_0(t)$. The subroutine **DepRound** (Gandhi et al., 2006) at step 12 draws M_t out of N items

with the specified marginal distribution $(\hat{\alpha}_1, \hat{\alpha}_2, ..., \hat{\alpha}_N)$, and is included in Appendix A.2.

5.5 Adaptive Learning of the Defender

In this section, we address the adaptive learning of the defender. Based on Algorithm 5.1 for the defender, the problem (5.4) can be recast by removing the constraint set, since will divide the problem to a scaling procedure and the MAB-VP. Formally, let $\mathbf{y}(t) := (y_k(t), \forall k \in \mathcal{N})$ for $t = 1, ..., T$ over a finite horizon T. For any search sequence of the defender $J = (J_t, t = 1, 2, ...)$ and a fixed sequence of attacks by the attacker $(\mathbf{y}(1), \mathbf{y}(2), ...)$, the total reward of the defender at T, denoted by $G^J(T)$, is given by

$$G^J(T) = \sum_{t=1}^{T} \sum_{j \in J_t} y_j(t) \tag{5.5}$$

Here, we obtain the maximum reward by consistently searching the subset $\mathcal{A}_{M_t}$, which is the most attacker-active location set at each time step t with cardinality M_t:

$$G_{\max}(T) = \max_{\mathcal{A}_{M_t}} \sum_{t=1}^{T} \sum_{k \in \mathcal{A}_{M_t}} y_k(t) \tag{5.6}$$

Let us define $\mathcal{A} = \cup_{M_t} \mathcal{A}_{M_t}$, where $\mathcal{A} \subset \mathcal{N}$. Note that if $M_t \in \{a, a+1, ..., b\}$, the location index subset $\mathcal{A}_{M_t}$ is defined such that $\mathcal{A}_a \subset \mathcal{A}_{a+1} \subset ... \subset \mathcal{A}_b = \mathcal{A}$.

The regret is then defined as

$$R(T) = G_{\max}(T) - G^J(T) \tag{5.7}$$

When $a = b$, i.e. M_t is time-invariant, the above regret reduces to the standard regret of MPMAB problem.

Since we care more about the competition against the optimal action in expectation, we define the pseudo-regret for our MAB-VP problem following the definition of pseudo-regret in (Bubeck et al., 2012) as:

$$\bar{R}(T) = G_{\max}(T) - \mathbb{E}[G^J(T)] \tag{5.8}$$

where the expectation is with respect to the randomness of the system state and the mixed-strategy of the defender.

We now give the first main theorem of this chapter, which bounds the expected weak regret of algorithm Exp3.M-VP.

Theorem 5.1. *For any $N > 0$ and for any $\eta \in (0, 1]$, if M_t is lower bounded and upper bounded by two positive integers a and b respectively, then*

$$\bar{R}_{\textit{Exp3.M-VP}}(T) = G_{\max}(T) - \mathbb{E}[G^J_{\textit{Exp3.M-VP}}(T)] \leq \left(1 + \frac{(e-2)b}{a}\right)\eta G_{\max}(T) + \frac{N}{\eta}\ln\frac{N}{b} \quad (5.9)$$

holds for any assignment of rewards and for any $T > 0$.

We present the proof of Theorem 5.1 as follows.

Proof. Let $W_t := \sum_{k=1}^{N} w_k(t)$ and $W'_t := \sum_{k=1}^{N} w'_k(t)$. Then, at each time step t,

$$\begin{aligned}
\frac{W_{t+1}}{W_t} &= \sum_{i\in S_0^c(t)} \frac{w_i(t+1)}{W_t} + \sum_{i\in S_0(t)} \frac{w_i(t+1)}{W_t} && (5.10a)\\
&= \sum_{i\in S_0^c(t)} \frac{w_i(t)}{W_t} \exp\left(\frac{\eta M_t}{N}\hat{y}_i(t)\right) + \sum_{i\in S_0(t)} \frac{w_i(t)}{W_t} && (5.10b)\\
&\leq \sum_{i\in S_0^c(t)} \frac{w_i(t)}{W_t}\left[1 + \frac{\eta M_t}{N}\hat{y}_i(t) + (e-2)\left(\frac{\eta M_t}{N}\hat{y}_i(t)\right)^2\right] + \sum_{i\in S_0(t)} \frac{w_i(t)}{W_t} && (5.10c)\\
&= 1 + \frac{W'_t}{W_t}\sum_{i\in S_0^c(t)} \frac{w_i(t)}{W'_t}\left[\frac{\eta M_t}{N}\hat{y}_i(t) + (e-2)\left(\frac{\eta M_t}{N}\hat{y}_i(t)\right)^2\right] && (5.10d)\\
&= 1 + \frac{W'_t}{W_t}\sum_{i\in S_0^c(t)} \frac{\frac{\hat{\alpha}_i(t)}{M_t} - \frac{\eta}{N}}{1-\eta}\left[\frac{\eta M_t}{N}\hat{y}_i(t) + (e-2)\left(\frac{\eta M_t}{N}\hat{y}_i(t)\right)^2\right] && (5.10e)\\
&\leq 1 + \frac{\eta}{(1-\eta)N}\sum_{i\in S_0^c(t)} \hat{\alpha}_i(t)\hat{y}_i(t) + \frac{(e-2)M_t\eta^2}{(1-\eta)N^2}\sum_{i\in S_0^c(t)} \hat{\alpha}_i(t)\hat{y}_i^2(t) && (5.10f)\\
&\leq 1 + \frac{\eta}{(1-\eta)N}\sum_{i\in J_t\cap S_0^c(t)} y_i(t) + \frac{(e-2)M_t\eta^2}{(1-\eta)N^2}\sum_{i\in\mathcal{N}} \hat{y}_i(t) && (5.10g)
\end{aligned}$$

Inequality (5.10c) uses $e^a \leq 1 + a + a^2$, $\forall a \in [0, 1]$, equality (5.10e) holds because of step 11 in Algorithm 5.1, inequality (5.10f) uses the fact that $\frac{W'_t}{W_t} \leq 1$, and the last inequality (5.10g) holds because $\hat{\alpha}_i(t)\hat{y}_i(t) = y_i(t) \leq 1$ for $i \in J_t$ and $\hat{\alpha}_i(t)\hat{y}_i(t) = 0$ for $i \notin J_t$. Then, according to inequality (5.10g) and by summing over t, we have

$$\begin{aligned}
\ln\frac{W_{T+1}}{W_1} &= \sum_{t=1}^{T} \ln\frac{W_{t+1}}{W_t} && (5.11a)\\
&\leq \sum_{t=1}^{T} \ln\left[1 + \frac{\eta}{(1-\eta)N}\sum_{i\in J_t\cap S_0^c(t)} y_i(t) + \frac{(e-2)M_t\eta^2}{(1-\eta)N^2}\sum_{i\in\mathcal{N}} \hat{y}_i(t)\right] && (5.11b)
\end{aligned}$$

$$\leq \frac{\eta}{(1-\eta)N} \sum_{t=1}^{T} \sum_{i \in J_t \cap S_0^c(t)} y_i(t) + \frac{(e-2)b\eta^2}{(1-\eta)N^2} \sum_{t=1}^{T} \sum_{i \in \mathcal{N}} \hat{y}_i(t) \tag{5.11c}$$

where inequality (5.11c) holds because $1 + y \leq e^y$ and $M_t \leq b$.

On the other hand, define $\mathcal{A}_b^*$ as the best location index subset with b elements.

Then,

$$\ln \frac{W_{T+1}}{W_1} \geq \ln \frac{\sum_{j \in \mathcal{A}_b^*} w_j(T+1)}{W_1} \tag{5.12a}$$

$$\geq \frac{\sum_{j \in \mathcal{A}_b^*} \ln w_j(T+1)}{b} - \ln \frac{N}{b} \tag{5.12b}$$

$$\geq \frac{\eta}{N} \sum_{j \in \mathcal{A}_b^*} \sum_{t: j \in S_0^c(t)} \hat{y}_j(t) - \ln \frac{N}{b} \tag{5.12c}$$

where inequality (5.12a) holds because $\mathcal{A}_b^* \subseteq \mathcal{N}$, inequality (5.12b) comes from the inequality of arithmetic and geometric means, i.e. $\frac{1}{b} \sum_{j=1}^{b} y_j \geq \left(\prod_{j=1}^{b} y_j \right)^{\frac{1}{b}}$, and inequality (5.12c) is obtained by recursively applying step 15 of Algorithm 1, which results in equality (5.13):

$$w_j(T+1) = \exp\left((b\eta/N) \sum_{t: j \in S_0^c(t)} \hat{y}_j(t) \right) \tag{5.13}$$

Note that we also have

$$\sum_{j \in \mathcal{A}_b^*} \sum_{t: j \in S_0(t)} \hat{y}_j(t) \leq \sum_{t=1}^{T} \sum_{i \in S_0(t)} y_j(t) \leq \frac{1}{1-\eta} \sum_{t=1}^{T} \sum_{i \in S_0(t)} y_j(t) \tag{5.14}$$

where the first equality is due to the fact that $\hat{y}_j(t) = y_j(t), \forall j \in S_0(t)$, and the last inequality holds because $\eta \in (0, 1]$.

Combining (5.11c), (5.12c), and (5.14), we have:

$$\sum_{j \in \mathcal{A}_b^*} \sum_{t: j \in S_0^c(t)} \hat{y}_j(t) + \sum_{j \in \mathcal{A}_b^*} \sum_{t: j \in S_0(t)} \hat{y}_j(t) - \frac{N}{\eta} \ln \frac{N}{b} \tag{5.15a}$$

$$\leq \frac{1}{(1-\eta)} G^J_{\text{Exp3.M-VP}}(T) + \frac{(e-2)\eta b}{(1-\eta)N} \sum_{t=1}^{T} \sum_{i \in \mathcal{N}} \hat{y}_i(t) \tag{5.15b}$$

Taking expectations of both sides of inequality (5.15), we obtain

$$\sum_{j \in \mathcal{A}_b^*} \sum_{t: j \in S_0^c(t)} \hat{y}_j(t) + \sum_{j \in \mathcal{A}_b^*} \sum_{t: j \in S_0(t)} \hat{y}_j(t) - \frac{N}{\eta} \ln \frac{N}{b} \tag{5.16a}$$

$$\leq \frac{1}{(1-\eta)}\mathbb{E}\left[G^J_{\text{Exp3.M-VP}}(T)\right] + \frac{(e-2)\eta b}{(1-\eta)N}\sum_{t=1}^{T}\sum_{i\in\mathcal{N}} y_i(t) \tag{5.16b}$$

$$\leq \frac{1}{(1-\eta)}\mathbb{E}\left[G^J_{\text{Exp3.M-VP}}(T)\right] + \frac{(e-2)\eta b}{(1-\eta)a}G_{\max}(T) \tag{5.16c}$$

where inequality (5.16b) uses the fact that $\mathbb{E}[\hat{y}_i(t)|S(1),...,S(t-1)] = y_i(t)$, and

$$\sum_{t=1}^{T}\sum_{i\in\mathcal{N}} y_i(t) \leq \frac{N}{a}G_{\max}(T) \tag{5.17}$$

Since $\mathcal{A}^*_b = \cup_{M_t}\mathcal{A}^*_{M_t}$ trivially holds, we have

$$G_{\max}(T) - \frac{N}{\eta}\ln\frac{N}{b} \leq \sum_{j\in\mathcal{A}^*_b}\sum_{t:j\in S^c_0(t)}\hat{y}_j(t) + \sum_{j\in\mathcal{A}^*_b}\sum_{t:j\in S_0(t)}\hat{y}_j(t) - \frac{N}{\eta}\ln\frac{N}{b} \tag{5.18}$$

Therefore, by combining (5.16) and (5.18), we obtain the inequality stated in the Theorem 5.1. □

By appropriately choosing the parameter η, we can obtain the following corollary:

Corollary 5.1. *Set* $\eta = \min\left\{1, \sqrt{\frac{Na\ln(N/b)}{(a+(e-2)b)bT}}\right\}$. *Then*

$$\bar{R}_{\textit{Exp3.M-VP}}(T) \leq 2\sqrt{\left(1+(e-2)\frac{b}{a}\right)}\sqrt{bTN\ln\frac{N}{b}}$$

holds for any $T > 0$ *and for any assignment of rewards.*

The proof of Corollary 5.1 follows the steps of the proof of Corollary 3.2 in (Auer et al., 2002b).

Proof. For any $T > 0$, we have $G_{\max}(T) \leq bT$. If $bT \leq \sqrt{\frac{Na\ln(N/b)}{a+(e-2)b}}$, then the bound is trivial since the expected regret cannot be more than bT. Otherwise, by Theorem 5.1, the expected regret is at most

$$2\sqrt{\left(1+(e-2)\frac{b}{a}\right)}\sqrt{bTN\ln\frac{N}{b}}$$

by plugging in $\eta = \sqrt{\frac{Na\ln(N/b)}{(a+(e-2)b)bT}}$. □

Note that when $a = b$, the upper bound in Corollary 5.1 is the same as the upper bound of Exp3.M in (Uchiya et al., 2010), and when $a = b = 1$ the upper bound becomes the same upper bound obtained for Exp3 in (Auer et al., 2002b).

Corollary 5.2. *Define* $\bar{s}_\infty := \liminf_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} s^J(t)\right]$ *as the average reward of the defender over infinite time horizon. Using the same parameter* η *as in Corollary 5.1, when the defender uses the Exp3.M-VP algorithm against the attacker who adopts a no-regret algorithm, we have* $\bar{s}_\infty = \frac{\nu}{N}$ *if* M_t *is a wide sense stationary process with mean* ν*.*

In order to prove Corollary 5.2, we need the following lemma, which was originally derived in (Wang & Liu, 2015).

Lemma 5.1. *When the defender (pursuer) is adopting Exp3.M and the attacker (evader) does not know the type of algorithm used by the adversarial opponent, then* $v = \frac{1}{N}$*, where* v *is the game value of the repeated constant-sum game for the defender.*

Then the proof of Corollary 5.2 is as follows.

Proof. The above problem is equivalent to the problem of two players playing an unknown repeated bimatrix game, where the game value $v_{i,t}$ ($i = 1, 2$ for the row and column player respectively) is changing over time. Define the game matrices as two $N \times N$ matrices $\mathbf{B}$ and $\mathbf{C}$, where $B_{ij} + C_{ij} = 1$ for any $(i, j) \in \mathcal{N} \times \mathcal{N}$. At each time t, the defender (i.e., the row player) chooses J_t rows of the matrix, and at the same time, the attacker (i.e., the column player) chooses exactly one column $I_t = k$. The defender then receives the payoff $\sum_{j\in J_t} B_{jk} = \sum_{j\in J_t} y_j(t)$. The defender uses a mixed strategy $\mathbf{p}_t$ at each time t, where $\mathbf{p}_t \in [0, 1]^N$, and the attacker chooses according to a probability vector $\mathbf{q}_t \in [0, 1]^N$. Note that the sum of $\mathbf{p}_t$ equals M_t and the sum of $\mathbf{q}_t$ equals 1. Let $v_{1,t}$ be the game value of the game matrix $\mathbf{B}$ at time t. Then by Corollary 5.1, we have

$$\mathbb{E}\left[\sum_{t=1}^{T}\sum_{j\in J_t} B_{jk}\right] = \mathbb{E}\left[\sum_{t=1}^{T}\sum_{j\in J_t} y_j(t)\right] \tag{5.19a}$$

$$\geq G_{\max}(T) - 2\sqrt{\left(1 + (e-2)\frac{b}{a}\right)} \times \sqrt{bTN\ln\frac{N}{b}} \tag{5.19b}$$

Let $\mathbf{p}_t$ be such that

$$v_{1,t} = \max_{\mathbf{p}_t}\min_{\mathbf{q}_t} \mathbf{p}_t^\intercal \mathbf{B}\mathbf{q}_t = \min_{\mathbf{q}_t}\max_{\mathbf{p}_t} \mathbf{p}_t^\intercal \mathbf{B}\mathbf{q}_t.$$

Then we have

$$G_{\max}(T) \geq \sum_{t=1}^{T}\sum_{i=1}^{N} p_{t,i} y_i(t) \tag{5.20a}$$

$$= \sum_{t=1}^{T} \mathbf{p}_t^\intercal \mathbf{y}(t) \tag{5.20b}$$

$$= \sum_{t=1}^{T} \mathbf{p}_t^\intercal \mathbf{B} \mathbf{q}_t \geq \sum_{t=1}^{T} v_{1,t} \tag{5.20c}$$

where $\mathbf{q}_t$ is a distribution vector whose I_t-th component is 1.

Combining (5.19) and (5.20), we have

$$E\left[\frac{1}{T}\sum_{t=1}^{T} s^J(t)\right] \geq \frac{1}{T}\sum_{t=1}^{T} v_{1,t} - 2\sqrt{\left(1+(e-2)\frac{b}{a}\right)} \times \sqrt{bN\ln\frac{N}{b}/T} \tag{5.21}$$

Note that at each time t, $v_{1,t} = M_t v_1$, where v_1 is the game value when the defender only chooses one location. Hence, by taking the limit of (5.21) and according to the law of large numbers we have

$$\bar{s}_\infty = \liminf_{T\to\infty} \frac{1}{T}\sum_{t=1}^{T} v_{1,t} = \nu v_1, \tag{5.22}$$

where the first equality comes from the fact that the attacker is also adopting a no-regret algorithm (e.g. Exp3). Finally, according to Lemma 5.1, we obtain the result. □

Corollary 5.3. *Under the setting that the defender adopts Exp3.M-VP and the attacker adopts a no-regret algorithm, assuming that M_t is a wide sense stationary process with mean ν, each player adopts the best response for the infinite-horizon problem.*

The proof can be obtained by extending the proof of the defender side in Corollary 5.2 to both sides, and is omitted for brevity. Note that in Corollary 5.2 and Corollary 5.3 we do not specify which type of learning algorithm the attacker is using, and the only assumption is that the attacker adopts a no-regret algorithm.

5.6 Adaptive Learning of the Attacker

We assume that the attacker adopts the Exp3 algorithm to randomly attack one location at each time step. The Exp3 algorithm runs the algorithm Hedge as a subroutine. Unlike the Hedge algorithm which directly takes advantage of the full information of the reward vector $\mathbf{x}(t) := (x_i(t), \forall i \in \mathcal{N})$, Exp3 observes partial information and feeds the simulated reward vector $\hat{\mathbf{x}}(t) := (\hat{x}_i(t), \forall i \in \mathcal{N})$ to the Hedge. The Hedge will then update $\hat{\beta}_i(t)$, which is the prediction of probability $\beta_i(t)$ for $i \in \mathcal{N}$. For more details about the Exp3 and Hedge algorithms, see Appendix A.1.

The defender adopts the Exp3.M-VP algorithm, which has a sublinear regret, as shown in Theorem 5.1. As a result, if the attacker favors one location, intuitively the defender will eventually identify this most attractive location, and fails to scan it only at a rate no more than sublinear in T. When M_t is a time-invariant constant, it follows immediately that the best strategy for the attacker

over an infinite time horizon is to treat each location equally, either in a stochastic or deterministic way. However, when M_t is a variable, the same argument cannot be trivially made.

Theorem 5.2. *Define* $\bar{r}_\infty := \liminf_{T\to\infty} \mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T} r^I(t)\right]$*, and let the location sequence* g *be the sequence of the greedy policy* π_{greedy}*, where* $g(t) = \arg\min_{i\in\mathcal{N}} \hat{\beta}_i(t)$ *for all* t*. If* M_t *is bounded by two positive integers* a, b *such that* $M_t \in \{a, a+1, ..., b\}$*, then under any policy* π *we have:*

$$\bar{r}_\infty \leq \frac{N-a}{N}$$

and under the greedy policy π_{greedy}*,*

$$\bar{r}_\infty \geq \frac{N-b}{N}$$

Proof. See Appendix A.3. □

Note that by Corollary 5.2, we can directly obtain the following result,

Corollary 5.4. *Under the setting that the defender adopts Exp3.M-VP, the attacker adopts Exp3, and* M_t *is a wide sense stationary process with mean* ν*, we have* $\bar{r}_\infty = \frac{N-\nu}{N}$*.*

Moreover, when M_t is a wide sense stationary process, following the proof of Theorem 5.2, it is not hard to show that even the greedy policy can obtain $\bar{r}_\infty = \frac{N-\nu}{N}$. Note that the above argument does not require Exp3.M-VP to have any property other than a no-regret guarantee, and therefore the greedy policy for the attacker can be a countermeasure against the entire family of no-regret algorithms. For the defender part, according to Corollary 5.2 and Corollary 5.4, a straightforward path to increase the average reward in an infinite time horizon is to increase the value of ν, i.e., assign more resources to the intrusion monitoring system.

5.7 Adaptive Adversarial Learning with Heterogeneous Rewards

In this section we consider heterogeneous rewards that are location-dependent. This corresponds to a more general setting, since in reality some locations (e.g., sensors) are more critical to the system than others. Let μ_k be the location-dependent reward corresponding to the k-th location. That is, the rewards of the attacker and the defender are $r^I(t) = \mu_{I_t} x_{I_t}(t)$ and $s^J(t) = \sum_{j\in J_t} \mu_j y_j(t)$, respectively. Without loss of generality, we assume that $\mu_1 \geq \mu_2 \geq ... \geq \mu_N$. We denote the frequency of location k being selected given the selection sequence I as $d_k^I(T)$ over a time horizon T, i.e.,

$$\frac{1}{T}\sum_{t=1}^{T} \mu_{I_t} = \frac{1}{T}\sum_{k=1}^{N} c_k^I(T)\mu_k = \sum_{k=1}^{N} d_k^I(T)\mu_k \tag{5.23}$$

where $c_k^I(T) = |\{t \leq T : I_t = k\}|$ and $d_k^I(T) = c_k^I(T)/T$. Note that $c_k^I(T)$ is the total number of times location k is selected by the attacker over horizon T given the selection sequence I.

Since the problem is no longer a constant-sum game under the setting of heterogeneous rewards, Corollary 5.3 and Corollary 5.4 cannot be directly applied. However, we can still show that when the reward for each location is heterogeneous, the average reward $\bar{r}_\infty$ in an infinite time horizon is bounded within an interval determined by a, b, and $\mu_k, k = 1, 2, ..., N$.

Theorem 5.3. *Given heterogeneous rewards, the average reward of the attacker $\bar{r}_\infty$ over an infinite time horizon is bounded within the interval $\left[\frac{K^*-b}{\sum_{k=1}^{K^*}\mu_k}, \frac{K^*-a}{\sum_{k=1}^{K^*}\mu_k}\right]$, where K^* is a constant determined by μ_k values such that $b \leq K^* \leq N$.*

In order to prove Theorem 5.3, we need Lemmas 5.2 and 5.3, as follows. Let $\text{supp}(\mathbf{d}) = \{k \in \mathcal{N} : d_k > 0\}$ for any feasible solution $\mathbf{d}$, and let K^* be the cardinality of $\text{supp}(\mathbf{d})$. Then we have the following lemmas:

Lemma 5.2. *For any optimal solution $\mathbf{d}^*$ of problem (5.27), (i) $\mu_k d_k^* = \mu_j d_j^*$ for any $k, j \in$ supp$(\mathbf{d}^*)$, and (ii), supp$(\mathbf{d}^*)$ consists of the indices of locations with the K^* highest μ.*

Lemma 5.3. *Problem (5.32) is lower bounded by $\frac{K^*-b}{\sum_{k=1}^{K^*}\mu_k}$.*

The proofs of Lemmas 5.2 and 5.3 can be found in the Appendices A.4 and A.5, respectively.

Now we shall give the proof of Theorem 5.3 as follows.

Proof. The average reward of the defender when using Exp3.M-VP is given by

$$E[G^J_{\text{Exp3.M-VP}}(T)] = E\left[\sum_{t=1}^{T}\sum_{j\in J_t}\mu_j y_j(t)\right] \tag{5.24a}$$

$$= \sum_{t=1}^{T}\sum_{k=1}^{N}\mu_k y_k(t)\beta_k(t) \tag{5.24b}$$

$$= \sum_{t=1}^{T}\mu_{I_t}\beta_{I_t}(t) \tag{5.24c}$$

$$= \sum_{t=1}^{T}\mu_{I_t} - E\left[\sum_{t=1}^{T} r^I(t)\right] \tag{5.24d}$$

for any realization I.

Then we have

$$\frac{1}{T}E\left[\sum_{t=1}^{T} r^I(t)\right] = \frac{1}{T}\sum_{t=1}^{T}\mu_{I_t} - \frac{1}{T}E[G^J_{\text{Exp3.M-VP}}(T)] \tag{5.25a}$$

$$\leq \sum_{k=1}^{N} \mu_k d_k^I(T) - \frac{1}{T}\left(G_{\max}(T) - 2\sqrt{\left(1+(e-2)\frac{b}{a}\right)}\sqrt{bTN\ln\frac{N}{b}}\right) \tag{5.25b}$$

$$\leq \sum_{k=1}^{N} \mu_k d_k^I(T) - \max_{J\in\mathcal{C}(\mathcal{N},a)} \sum_{j\in J} \mu_j d_j^I(T) + 2\sqrt{\left(1+(e-2)\frac{b}{a}\right)}\sqrt{bN\ln\frac{N}{b}/T} \tag{5.25c}$$

where $\mathcal{C}(\mathcal{N},a) = \{\mathcal{S}\subseteq\mathcal{N} : |\mathcal{S}| = a\}$, namely, the set of all subsets of size a in $\mathcal{N}$. The second inequality uses the fact that

$$\begin{aligned} G_{\max}(T) &\geq \max_{J\in\mathcal{C}(\mathcal{N},a)} \sum_{t=1}^{T}\sum_{j\in J} \mu_j y_j(t) \\ &= \max_{J\in\mathcal{C}(\mathcal{N},a)} \sum_{j\in J} \mu_j c_j^I(T) \end{aligned}$$

Therefore, by having T approach infinity, we have

$$\bar{r}_\infty \leq \liminf_{T\to\infty} E\left[\sum_{k=1}^{N} \mu_k d_k^I(T) - \max_{J\in\mathcal{C}(\mathcal{N},a)} \sum_{j\in J} \mu_j d_j^I(T)\right] \tag{5.26}$$

for any policy π.

Consider the following optimization problem

$$\underset{\mathbf{d}\in\Delta_N}{\text{maximize}} \sum_{k=1}^{N} \mu_k d_k - \max_{J\in\mathcal{C}(\mathcal{N},a)} \sum_{j\in J} \mu_j d_j \tag{5.27}$$

where Δ_N is the set of distributions over $\mathcal{N}$ and $\mathbf{d} = (d_k, k\in\mathcal{N})$. Let the optimal solution and its objective function value be $\mathbf{d}^*$ and $r_{\max}$, respectively. Then we have

$$\bar{r}_\infty \leq r_{\max} = \sum_{k=1}^{N} \mu_k d_k^* - \max_{J\in\mathcal{C}(\mathcal{N},a)} \sum_{j\in J} \mu_j d_j^* \tag{5.28}$$

Without loss of generality, we assume that $supp(\mathbf{d}^*) = \{1,2,...,K^*\}$. Therefore, according to Lemma 5.2, we have $d_k^* = \frac{1/\mu_k}{\sum_{j=1}^{K^*} 1/\mu_j}$ for all $k \leq K^*$. Then the optimal value of problem (5.27) is given by $(K^*-a)/\sum_{j=1}^{K^*} 1/\mu_j$, which is increasing with respect to the value of $K^* = 1,2,...,N$. This gives the upper bound of $\bar{r}_\infty$.

When the defender adopts Exp3M-VP, we have

$$E[G^I_{\text{Exp3}}(T)] \geq G'_{\max(T)} - o(T) \tag{5.29}$$

where $G^I_{\text{Exp3}}(T)$ is the total reward of the attacker when adopting Exp3, and $G'_{\max}(T) = \max_{k\in\mathcal{N}} \sum_{t=1}^{T} x_k(t)$ is the maximum total reward the attacker can gain when selecting a fixed location to attack.

Similarly, define $h^J_k(T) = |\{t \leq T : k \in J_t\}|$ and $l^J_k(T) = h^J_k(T)/T$. Then, we have

$$G'_{\max}(T) = \max_{k\in\mathcal{N}} \mu_k(T - h^J_k(T)) \tag{5.30}$$

Thus, the average reward $\bar{r}_\infty$ of the attacker over an infinite time horizon is lower bounded by

$$\bar{r}_\infty \geq \liminf_{T\to\infty} E\left\{\max_{k\in\mathcal{N}} \mu_k(1 - l^J_k(T))\right\} \tag{5.31}$$

Consider the following optimization problem

$$\underset{c\in\Delta_N}{\text{minimize}} \max_{k\in\mathcal{N}} \mu_k(1 - l_k) \tag{5.32}$$

and denote the optimal value of problem (5.32) as $r_{\min}$. Then according to Lemma 5.3, $r_{\min} = \frac{K^*-b}{\sum_{k=1}^{K^*}\mu_k}$, which gives us the lower bound of $\bar{r}_\infty$. □

Theorem 5.3 when the attack success rate of the attacker is not 100 percent for all locations, where μ_k represents the success rate of attacks on location k for the attacker. Note that although Theorem 5.3 assumes heterogeneous rewards, it can be simply applied to homogeneous rewards as well. Figure 5.1 shows the range for the attacker's average reward in an infinite time horizon under different attack success rates, where we assume the same attack success rate for all locations for simpler visualization. Note that we do not even assume that M_t is a wide sense stationary process; the only assumption here is that it is confined within a range with lower and upper bounds a and b, respectively. The shaded blue region in Figure 5.1 indicates the potential reward the attacker can obtain in infinite time, and the red and blue lines indicate the lower and upper bounds on the attacker's average reward in infinite time, according to Theorem 5.3. When the attack success rate is 1, the lower and upper bounds become equivalent to the bounds in Theorem 5.2. It is straightforward to see that the lower the success rate of the attack, the safer the system will be.

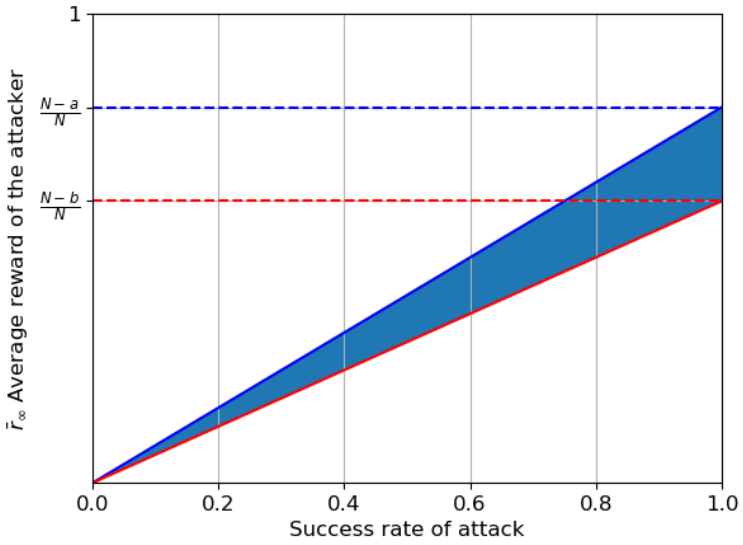


Figure 5.1: Range of the average reward of the attacker in an infinite time horizon under different attack success rates.

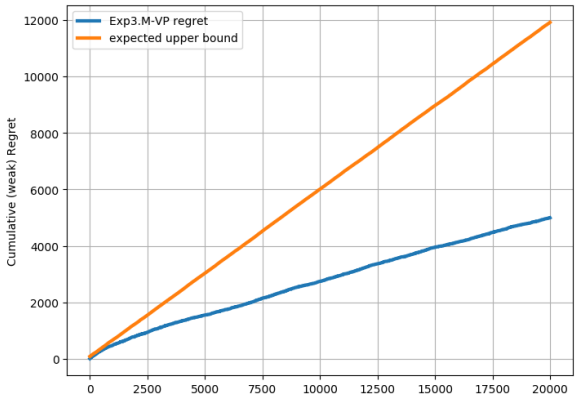
5.8 Numerical Experiment

We conducted extensive simulations illustrating the performance of the proposed algorithm and policy. Our numerical analysis consists of three parts. In section 5.8.1, we conduct simulations to test the Exp3.M-VP performance under a single-player setting. In section 5.8.2, we compare the performance of Exp3.M-VP with several bandit learning algorithms, i.e., the Exp3, Exp3.M, upper-confidence Bound (UCB) (Auer et al., 2002a), and ϵ-greedy algorithms (Sutton et al., 1998), on real in-vehicle network datasets from the Car-Hacking datasets (Seo et al., 2018). In section 5.8.3, we run simulations on the proposed game model and algorithmic solutions.

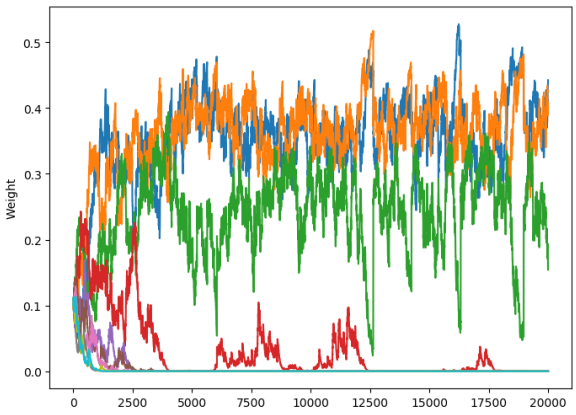
5.8.1 Simulations on a Single Player

In this section we consider the single-player setting, where the Exp3.M-VP algorithm was evaluated on a ten-armed bandit problem with rewards for arms drawn independently from Bernoulli distributions with means $\{0.75, ..., \frac{3}{4k}, ..., 0.075\}$, with $k = 1, 2, ..., 10$. This scenario was simulated over a fixed time horizon $T = 20,000$ time steps. The number of arms played at each time step is drawn independently from a discrete uniform distribution over $\{1, 2, 3\}$. Parameter η is set to 0.1.

Figure 5.2(a) shows the regret of Exp3.M-VP versus the expected upper bound of the regret from Theorem 5.1. We can see that the actual regret of Exp3.M-VP has a smaller rate than its expected upper bound and the discrepancy becomes larger as time increases. Figure 5.2(b) shows



(a) Exp3.M-VP regret (blue curve) and expected upper bound of regret (orange curve).



(b) Normalized weights of 10 arms over 20,000 time steps.

Figure 5.2: Simulation of Exp3.M-VP on a ten-armed bandit problem.

the change of the normalized weight for each location over the entire time horizon. As shown in this figure, Exp3.M-VP chooses the top three locations (i.e. the blue, orange, and green curves) with the highest average reward only after a short period of time, and the rest of weights vanish to nearly 0. The reason why only three locations pop up is that M_t, i.e. the number of the arms played at each time, is within the set $\{1, 2, 3\}$. The fluctuations of the weights are partly due to the fact that the Exp3.M-VP algorithm needs to explore different locations in order to update the choice prediction and estimation, and partly due to the fact that the sum of the weights must always equal to M_t, which is changing over time.

5.8.2 Evaluations on Car-Hacking Dataset for the Defender

In this section we compare Exp3.M-VP with Exp3, Exp3.M, UCB, and the ϵ-greedy algorithms by implementing these algorithms over two in-vehicle network datasets from the Car-Hacking datasets. The Car-Hacking datasets are generated by logging the Controller Area Network (CAN) traffic via the OBD-II port from a real vehicle while message injection attacks were made. The Datasets each contain 300 intrusions of message injections over 26 unique CAN IDs. Each intrusion is performed for 3 to 5 seconds, and each dataset has a total of 30 to 40 minutes of the CAN traffic. Specifically, we test the performance on the spoofing attack datasets, which were conducted on the RPM gauze and the driving gear. That is, among 26 arms representing CAN IDs, two of them (RPM gauze and driving gear) contained spoofing attacks.

Figure 5.3 shows the cumulative average rewards for each bandit learning algorithm used by the defender. The experiments were conducted over $T = 7,000$ time steps, and the number of arms played by Exp3.M-VP was sampled from a truncated Gaussian distribution within the interval [1,3],

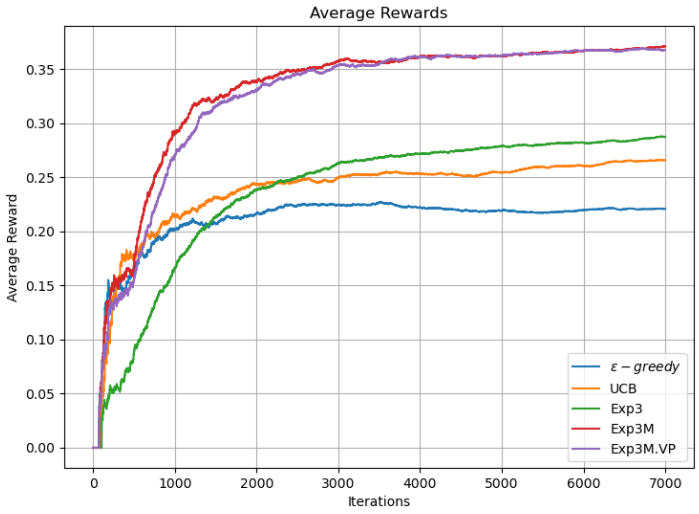


Figure 5.3: Cumulative average rewards for ϵ-greedy, UCB, Exp3, Exp3.M, and Exp3.M-VP.

with mean 2 and standard deviation 0.8. The number of arms played by Exp3.M was set to 3. We can see that both Exp3.M and Exp3.M-VP obtain higher cumulative average rewards than other single-play setting algorithms, due to the benefits from multiple or variable plays. Exp3.M-VP in this setting is a constrained version of Exp3.M, since the number of arms in Exp3.M (i.e., 3) is an upper bound on the number of arms available to Exp3.M-VP (i.e., $[1, 3]$). This indicates that Exp3.M-VP may have access to a smaller number of arms due to resource constraints. To make it more challenging, Exp3.M-VP does not know in advance the number of number of arms it may have access to in the future. Therefore, not surprisingly, Exp3.M obtains a slightly higher cumulative average reward than Exp3.M-VP. However, interestingly, eventually the cumulative average rewards of Exp3.M and Exp3.M-VP approach the same value. This demonstrates the power of the Exp3.M-VP algorithm: despite the fact that in average Exp3.M-VP plays fewer arms than Exp3.M, it can match the performance of Exp3.M. The reason is that only 2 out of of 26 CAN-IDs contained spoofing attacks, and after a period of time (i.e., around 3500 iterations), both Exp3.M and Exp3.M-VP are able to identify the top two most rewarded CAN-IDs.

We further conduct sensitivity analysis on the number of arms played by Exp3.M and Exp3.M-VP. Specifically, we test the performance of the two algorithms with $M = \nu \in \{4, 5, 6, 7\}$, where M is the number of arms played by Exp3.M. For each ν, we sample M_t from a truncated Gaussian distribution within the interval $[\nu - 1, \nu + 1]$, with mean ν and standard deviation 0.8. As such, in this set of experiments the number of arms played by Exp3.M is the mean value of the number of arms played by Exp3.M-VP. Figure 5.4 shows the results. This figure demonstrates the average reward of the two algorithms under four values for M and ν. We can see that the performance

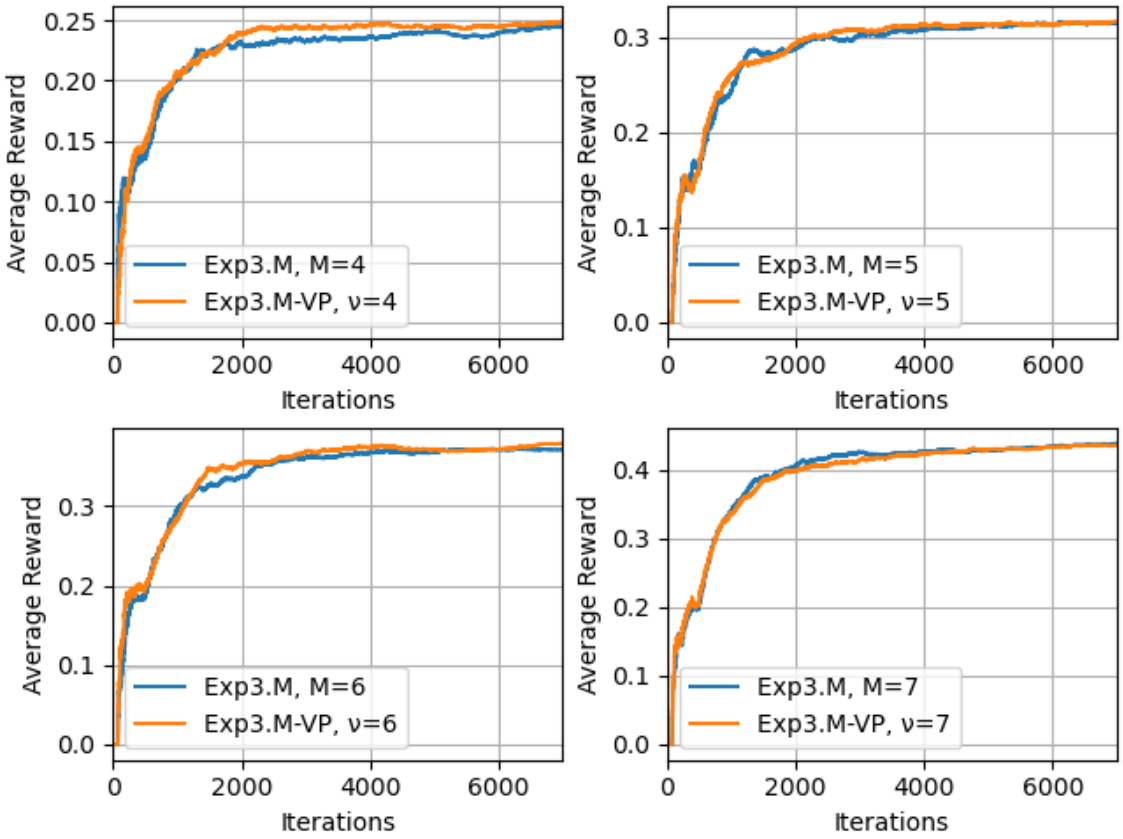


Figure 5.4: Average reward of Exp3.M and Exp3.M-VP under different M_t and ν.

of the two algorithms are very close, mainly due to the fact that $M = \nu$. Note that here, in some instances Exp.M-VP will have access to less resources/arms, and in some instances more. As a result, throughout the iterations, sometimes Exp3.M outperforms Exp3.M-VP, and sometimes it underperforms. However, eventually both algorithms reach the same reward and successfully identify the attacked arms. This again demonstrates the strength of Exp3.M-VP, because the number of arms are determined exogenously and therefore Exp3.M-VP is able to match the reward obtained by Exp3.M under uncertainly on the number of available arms at each time.

5.8.3 Simulations on Two Players

We now consider a game setting where two players, i.e., an attacker and a defender, are playing the pursuit-evasion game against each other. This corresponds to the realistic scenario where a malicious hacker is trying to compromise either the sensor/ECU in an in-vehicle sensor network, or the entire vehicle/infrastructure in an interconnected transportation system without being identified by the intrusion monitoring system. At the same time, the intrusion monitoring system is trying to identify as many compromised locations as possible to minimize the potential loss. We consider a

ten-armed bandit problem for the two players, where the attacker adopts Exp3 and the defender adopts Exp3.M-VP. The scenario was simulated over $T = 100,000$ time steps, and the number of arms played by the defender was sampled from a truncated Gaussian distribution within the interval $[1, 3]$, with mean 2 and standard deviation 0.8. The parameter η for both Exp3 and Exp3.M-VP was set according to Corollary 5.1.

Figure 5.5 illustrates the average reward and the equilibrium reward for the two players. Since we have $N = 10$ and $\nu = 2$, according to Corollary 5.2 and Corollary 5.4, the equilibrium rewards for the attacker and the defender are 0.8 and 0.2, respectively. We can see that the average rewards of both players converge to the equilibrium rewards after a relatively short period, and after that the average rewards stay around the equilibrium reward with small fluctuations. The fluctuations are due to the fact the Exp3 and Exp3.M-VP use randomized policies and need to occasionally explore different locations in order to update the choice predictions and estimations.

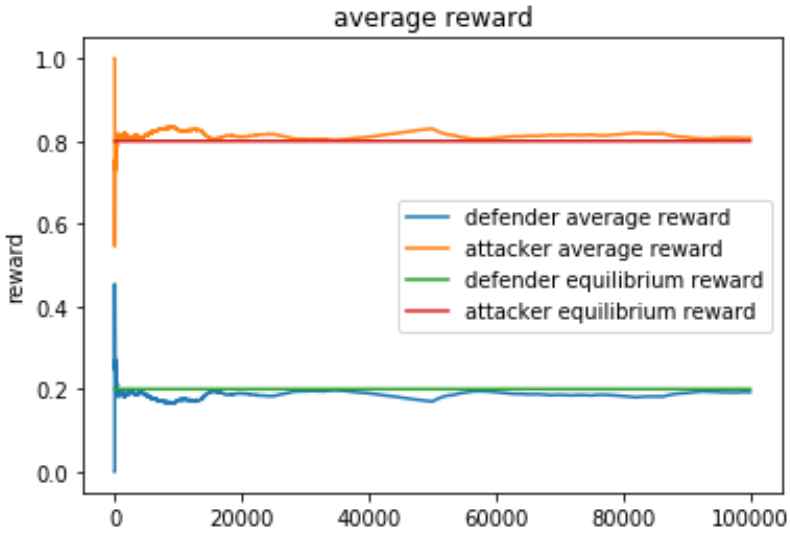


Figure 5.5: Average reward of the attacker and the defender over 100,000 time steps.

5.9 Conclusion

In this chapter, we extend the adversarial/non-stochastic MPMAB to the case where the number of plays can change in time, and propose the Exp3.M-VP algorithm for obtaining the variable-play property. This extension is motivated by the uncertainty of resources allocated to the intrusion monitoring system to scan at each time in resource-constrained systems, such as an interconnected transportation system. We derive a sublinear regret bound for Exp3.M-VP, which simplifies to the existing bounds in the literature when the number of arms played at each time is constant. We

introduce a game setting where an attacker and a defender play a pursuit-evasion game against each other. The defender, who represents the intrusion monitoring system, adopts Exp3.M-VP and the attacker, who represents the malicious hacker, adopts Exp3. We derive the condition under which a Nash equilibrium of the strategic game exists. Finally, we consider heterogeneous rewards for arms, and obtain lower and upper bounds on the average rewards for the attacker in an infinite time horizon. We provide several numerical experiments that demonstrate our results.

This work provides insights on deploying an intrusion monitoring system either in an in-vehicle network or a transportation network: In order to minimize the potential loss of the system from cyber threats, one can either increase the average resources allocated to intrusion monitoring, or change the potential reward vector for each location to reduce the reward bound in Theorem 5.3. One of the potential extensions of this work is to consider the connectivity or correlations between different arms, which can take into account the spread of the cyber attacks, and use such information to facilitate the decision making of the intrusion monitoring system.

CHAPTER 6

Dynamic Security Resource Allocation for Connected and Automated Vehicles

6.1 Introduction

Recent developments in connected and automated vehicle (CAV) technology bring with them the promise of increased safety and reduced vehicles crashes by as much as 80%, while reducing by 6.9 billion hours the time Americans spend in traffic annually (USDOT, 2016).

CAVs require full and reliable coverage of their surrounding environment, which necessitates sophisticated onboard hardware and software for communications and autonomous driving than a human-driven vehicle. For example, perception in CAVs may be achieved using several sensor systems, including cameras, LiDAR, and radar. Once a CAV collects disparate sources of sensor measurements, it typically fuses sensor data to create more coherent, reliable, and precise pieces of information than would be possible to obtain from each of these sources individually (Kocić et al., 2018).

On the other hand, the extensive sensor systems and the communication channel serve as attack surfaces in CAVs, giving rise to cybersecurity concerns (Petit & Shladover, 2014). As mentioned in Chapter 5, it is imperative to allocate security monitoring resources to CAVs in order to detect potential cyberattacks. However, all the aforementioned operations, including sensing, information fusion, and security monitoring tasks require energy from the limited energy supply aboard of a CAV. These energy requirements are in addition to the energy used for vehicle propulsion and computational resources for decision making. As the energy required for driving changes under different traffic conditions and based on the sophistication of the driving environment, the amount of energy that can be dedicated to threat monitoring may not be deterministic. Furthermore, it may not be realistic to continuously monitor all sensors, mainly due to limited supply of energy at the disposal of the vehicle. In general, there exists a trade-off between the desired outcomes (e.g., attack detection sensitivity) and the computational ability and/or energy resources. Thus, security resources should be dynamically allocated depending not only on the security state of the CAV

but also the energy required for other driving tasks. Moreover, the actual security state of a CAV in practice may not be fully observable as there is a chance of not detecting an attack even when security resources are allocated for sensor monitoring.

In this chapter, we address the dynamic security resource allocation for a CAV through sequential decision making under uncertainty and partial information. Specifically, in this chapter we consider a CAV driving on a planned route and subject to potential cyberattacks. The true status of cyberattacks can only be observed via an attack detection monitor deployed on the CAV or on the cloud. The amount of security resources that can be allocated for monitoring depends on the energy supply of the CAV. At each time epoch, the CAV decides the amount of security resource to be allocated for attack detection. We assume that the detection recall/sensitivity is a function of the amount of allocated security resource. We further associate costs to undetected attacks and an unfinished trip due to energy depletion. As such, there is a trade-off between monitoring the CAV system to improve the detection sensitivity, and the risk of being unable to finish the trip. We develop a partially observable Markov decision process (POMDP) model that captures this trade-off by prescribing a security resource allocation policy to minimize the total discounted cost of a CAV trip over a finite horizon. To the best of our knowledge, it is the first study addressing the aforementioned trade-off and devising a dynamic security resource allocation policy for CAVs. The work in this chapter can be regarded as a complementary supplement to Chapter 5, where the proposed POMDP determines the number of sensors to be scanned for the Exp3.M-VP algorithm (denoted by M_t in that chapter), and the Exp3.M-VP returns the real-time detection results to POMDP. Using this feedback, POMDP can decide the number of security resources to allocate to the threat monitoring system in the next time epoch.

The remainder of the chapter is organized as follows. We first review the related work in dynamic resource allocation (DRA) in section 6.2. Then, in section 6.3 we provide the mathematical model of our POMDP. In section 6.4 we present the results of numerical experiments for a CAV with three candidate sensors. Finally, we conclude the chapter in section 6.5.

6.2 Literature Review

The dynamic resource allocation (DRA) problem is one of the most challenging problems in the family of resource management problems. In general, DRA is widely used in applications where a pool of resources can be accessed in wired or wireless networks. DRA seeks to dynamically assign the resources needed for different types of services. In the past decades, DRA has attracted extensive attention in the field of cloud computing (Chandra et al., 2003; Xiao et al., 2013), wireless communications (Gunduz & Erkip, 2007; Zhang et al., 2010), and computer systems (Singh et al., 2017). Resource allocation becomes critical when there are limited resources available in the

presence of high demand. At the same time, DRA should be able to handle less critical scenarios with low demand and high resource availability.

The solution methodologies proposed for the DRA problem vary depending on the application. For instance, Zhang et al. (2010) formulated and solved DRA in cognitive radio networks as a convex optimization problem. Vengerov (2007) proposed a reinforcement learning in conjunction with a fuzzy rulebases framework to solve the DRA problem in distributed computing systems. Among solution methodologies developed for the DRA problem, the Markov decision process (MDP) has attracted high interest. For example, Tang et al. (2018) addressed the adaptive virtual resource allocation in 5G networks by formulating a constrained MDP model. In (Oddi et al., 2013), the authors proposed a multi-cloud resource allocation algorithm based on MDP to dynamically assign the resources to a set of IT requests.

Although a moderate volume of research has been conducted in the filed of DRA, it is not yet widely explored in the field of cybersecurity. One of the most related studies is (Njilla et al., 2017), which considered the setting where the cybersecurity resources are divided into two layers, namely, agility and recovery. This study developed an MDP framework for cybersecurity resource allocation. There is still a gap in the literature for security resource allocation in CAVs. Nayak et al. (2022) conducted a survey on resource allocation, security, and data privacy of autonomous vehicles, but they did not explicitly consider security resource allocation.

Besides the aforementioned studies using MDP models, Bayesian frameworks, including both MDP and POMDP, have been used in the context of CAV decision-making (van Wyk et al., 2020; Liu et al., 2022) and cybersecurity (Tipireddy et al., 2017). For instance, van Wyk et al. (2020) studied the control switching problem between the human driver and the automated control entity in semi-autonomous vehicles, by developing an MDP model and a POMDP model to prescribe the optimal switching policy. Liu et al. (2022) developed a locally-optimal motion planner with an MDP model that captures network-level information. Tipireddy et al. (2017) proposed an agent-centric approach for a general cybersecurity decision-support problem using POMDP models, which is one of few studies using POMDP in the context of cybersecurity.

6.3 Model Formulation

Consider a CAV driving on a planned route. At each time epoch, the CAV needs to determine the number of security resource to allocate for attack detection, while at the same time ensuring that it can complete its trip. The more security resource allocated, the less likely it would be for an attacker to successfully comprise the system. Denote by $t \in \mathcal{T} \triangleq \{0, 1, 2, ..., T\}$ the time epoch, where T is the number of time epochs in the planning horizon. Denote by $x_t \in \mathcal{X} \triangleq \{0, 1\}$ the attack state of the CAV at time epoch t, representing whether the CAV is under attack ($x_t = 1$) or not ($x_t = 0$). We

follow the same setting from Chapter 5 where at each time epoch the attacker selects one sensor to attack. As long as an attack occurs at time epoch t, we have $x_t = 1$. At each time epoch t, we denote by $e_t \in \mathcal{E} \triangleq \{1, ..., E\}$ the number of units of energy required for driving/propulsion, where E is the initial number of energy units available to the CAV, and by $r_t \in \mathcal{E}$ the CAV's remaining energy. The state vector of the CAV is then defined as a three dimensional tuple $\boldsymbol{s_t} \triangleq (x_t, e_t, r_t) \in \mathcal{S}$, where $\mathcal{S} \triangleq \mathcal{X} \times \mathcal{E} \times \mathcal{E}$ represents the state space.

Note that we assume that at each time epoch t the system state x_t, i.e., whether the CAV is under attack or not, is not outwardly observable and can only be partially observed from the attack detection monitor (e.g., Exp3.M-VP in Algorithm 5.1). Instead, the observation vector $\boldsymbol{o_t}$ is obtained by the CAV at time t, which is defined as $\boldsymbol{o_t} \triangleq (y_t, e_t, r_t) \in \mathcal{O} \triangleq \mathcal{Y} \times \mathcal{E} \times \mathcal{E}$. At each time epoch t, the attack detection monitor returns the detection result $y_t \in \mathcal{Y} \triangleq \{0, 1\}$ on the CAV system, where $y_t = 1$ indicates a detected attack at the current time epoch t, and 0 otherwise. Then, the system belief state, i.e., the probability that the system is under the state $x \in \mathcal{X}$ at time epoch t, is updated based on this observation. Let $\boldsymbol{\pi_t} \triangleq [\pi_t(0), \pi_t(1)]$ denote the vector of belief state at time epoch t, where $\pi_t(x)$ is the probability that the system is in core state x at the beginning of time epoch t. We have $\sum_{x \in \mathcal{X}} \pi_t(x) = 1$. Note that we omit the states of $e \in \mathcal{E}$ and $r \in \mathcal{E}$ as they are fully observable.

Assume there are N individual candidate sensors in the CAV, all eligible for security monitoring. Denote $a_t \in \mathcal{A}$ as the action representing the units of security resources allocated at time t, where $\mathcal{A} \triangleq \{1, ..., N\}$ is the action set. We assume that monitoring each sensor consumes one unit of energy. Once the CAV allocates the security resources to the security monitor, we assume that it adopts a separate algorithm to determine which subset of sensors to be scanned at time epoch t, and returns detection results to POMDP.

Let $\boldsymbol{\Psi_a}$ denote the information matrix associated with action $a \in \mathcal{A}$. Specifically, $\Psi_a(y|x)$ denotes the probability that the observed system state is anomalous/non-anomalous (i.e., $y = 1$ or $y = 0$, respectively), given the core state $x \in \mathcal{X}$ and action a. Note that again we omit the states components $e \in \mathcal{E}$ and $r \in \mathcal{E}$ as they are fully observable.

Let $\mathbb{P}_t$ denote the one-step transition probability matrix (TPM) of the system at time epoch t, where $\mathbb{P}_t(x', e'|x, e)$ denotes the state-transition probability at time epoch t from state x and e to state x' and e'. When the state variable e is independent from x, the state transition probability $\mathbb{P}_t(x', e'|x, e)$ can be decomposed as $\mathbb{P}_t(x', e'|x, e) = \mathbb{P}_t(x'|x) \cdot \mathbb{P}_t(e'|e)$. Note that the state-transition probability can be obtained from historical data. As discussed, the core state x can be only observed partially. Therefore, at each decision epoch, depending on the action taken, the system belief state can be updated by Bayes' rule using the observation vector $\boldsymbol{o_t}$. Let $\boldsymbol{\pi_{t+1}} = \beta[\boldsymbol{\pi_t}, a_t, \boldsymbol{o_t}]$ denote the vector of belief state at time $t + 1$ starting from the system belief $\boldsymbol{\pi_t}$ at time t, under action a_t, and the observed system state $\boldsymbol{o_t} = (y_t, e_t, r_t)$, where $\beta[\cdot]$ is the hidden Markov model (HMM) filter for estimating the underlying state of a Markov chain given noisy observations. In the

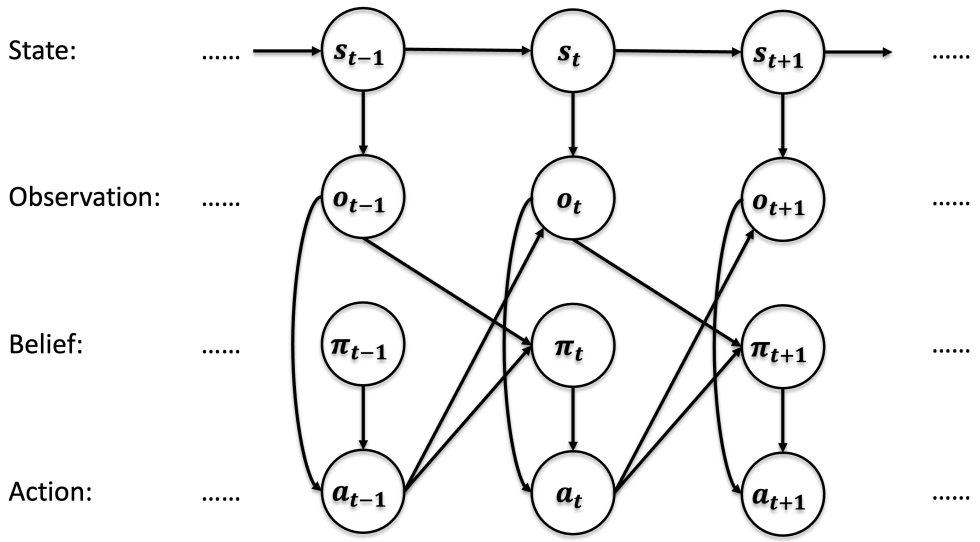


Figure 6.1: POMDP diagram for the security resource allocation problem.

next time epoch, the state belief is updated to obtain $\pi_{t+1}(x)$. Specifically, the belief about the core state $x' \in \mathcal{X}$ at time $t+1$ is updated as follows:

$$\pi_{t+1}(x') = \beta[\boldsymbol{\pi_t}, a_t, \boldsymbol{o_t}](x') \triangleq \frac{\Psi_{a_t}(y_t|x') \sum\limits_{x\in\mathcal{X}} \mathbb{P}_t(x'|x) \cdot \pi_t(x)}{\sum\limits_{x'\in\mathcal{X}} \Psi_{a_t}(y_t|x') \sum\limits_{x\in\mathcal{X}} \mathbb{P}_t(x'|x) \cdot \pi_t(x)} \tag{6.1}$$

Figure 6.1 presents the flow diagram of the proposed POMDP model for the security resource allocation problem. For this problem, the state vector $\boldsymbol{s_t}$ evolves according to the TPM. At each time epoch t, the system obtains the observation vector $\boldsymbol{o}_t$. Based on the observation vector and the belief vector $\boldsymbol{\pi_t}$ at the current time, POMDP chooses the action a_t based under a specific policy. After taking the action, an immediate cost occurs. Then the action a_t and the observation vector $\boldsymbol{o}_t$ are used to update the belief vector $\boldsymbol{\pi_{t+1}}$ for the next time epoch using equation (6.1). Note that the action a_t at time t will also affect the observation vector obtained at time epoch $t+1$ as the information matrix is action-dependent.

The objective of the POMDP model is to minimize the total expected discounted cost over a finite horizon. Denote λ as the discount factor and $V_t(\boldsymbol{\pi_t}, e_t, r_t)$ as the value function representing the total discounted cost-to-go starting at time t with the belief $\boldsymbol{\pi_t}$ of attack state x, with the required energy e_t and the remaining energy r_t. Also, denote $V_t^*(\boldsymbol{\pi_t}, e_t, r_t)$ as the minimum total expected discounted cost-to-go starting at time t with the belief $\boldsymbol{\pi_t}$ of attack state x, with the required energy e_t and the remaining energy r_t. As we have a finite horizon, the boundary condition of the value

function is defined as follows:

$$\begin{cases} V_T^*(\boldsymbol{\pi_T}, e_T, r_T) = -c_R \cdot r_T & \forall e_T \in \mathcal{E},\ \text{if } r_T \geq 0 \\ V_t^*(\boldsymbol{\pi_t}, e_t, r_t) = c_U & \forall e_t \in \mathcal{E},\ \forall t \in \mathcal{T},\ \text{if } r_t < 0 \end{cases} \tag{6.2}$$

where c_R represents the unit reward for the remaining energy when finishing the trip, and c_U represents the termination cost induced by an unfinished trip. The boundary conditions (6.2) include two stopping criteria: (i) if at the end of the horizon T, the remaining energy r of the CAV is non-negative, the CAV receives a reward of c_R for every unit of remaining energy; Or (ii) if the CAV runs out of energy before time T, then it incurs a one time cost of c_U for not completing the trip.

Then, we can obtain the Bellman equation of $V_t^*(\boldsymbol{\pi_t}, e_t, r_t)$ as follows:

$$\begin{aligned} V_t^*(\boldsymbol{\pi_t}, e_t, r_t) = \min_{a \in \mathcal{A}} \sum_{x \in \mathcal{X}} \pi_t(x) \Bigg[\bar{C}(x,a) + \gamma \sum_{\substack{y \in \mathcal{Y} \\ \boldsymbol{o}=(y,e_t,r_t)}} \sum_{\substack{x' \in \mathcal{X} \\ e' \in \mathcal{E}}} \Psi_a(y|x') \\ \cdot \mathbb{P}_t(x', e'|x, e_t) \cdot V_{t+1}^*(\beta[\boldsymbol{\pi_t}, a, \boldsymbol{o}], e', r_t - e_t - a) \Bigg] \end{aligned} \tag{6.3}$$

where $\bar{C}(x,a) = \mathbb{E}_t[C_t(x,a)]$ represents the expected immediate reward function of the attacker over time, which is also the expected immediate cost function of the CAV. $C_t(1,a) = c_A$ if the security monitor fails to detect an existing attack, and 0 otherwise, where c_A is a constant cost for every undetected attack. $C_t(0,a) = 0$ for any a as we do not penalize the CAV if there is no attack. Equation (6.3) calculates the minimum total expected discounted cost-to-go starting at time t with belief $\boldsymbol{\pi_t}$, required energy e_t, and remaining energy r_t.

6.4 Numerical Experiments

In this section, we illustrate our proposed POMDP model by solving a small-sized problem. We consider a CAV with three candidate sensors, subject to cyberattacks. We adopt the setting from Chapter 5 where the security monitor uses Exp3.M-VP algorithm. Specifically, we consider the scenario in Figure 6.2, where at each time t, POMDP determines a_t units of security resources for Exp3.M-VP. The Exp3.M-VP algorithm then select a_t sensors among three sensors and detects potential cyberattacks. We consider an imperfect detector where false positives and false negatives may occur. The Exp3.M-VP algorithm then return its detection results y_t as a part of the observation $\boldsymbol{o_t}$ vector, and POMDP updates its belief vector using equation (6.1) based on this observation.

The CAV system is initialized with a total of 12 units of energy, i.e., $E = 12$, and the trip horizon $T = 4$. The discount factor is set to $\lambda = 0.95$. In this small-sized problem we assume

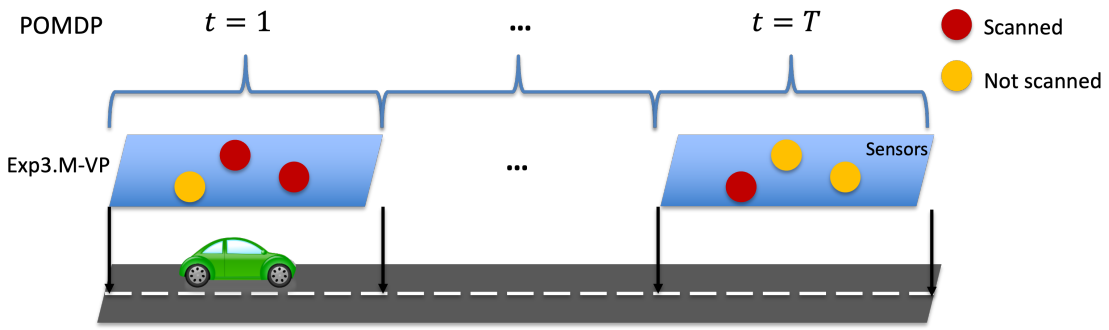


Figure 6.2: Collaboration of POMDP and Exp3.M-VP.

the required energy for driving at each time is given as a prior, specified as $[1, 3, 1, 3]$ in this case. Attacks occur according to the TPM as follows,

$$\mathbb{P}_t = \begin{bmatrix} 0.6 & 0.4 \\ 0.2 & 0.8 \end{bmatrix}, \forall t \in \mathcal{T} \tag{6.4}$$

Note that we decrease the dimension of the TPM to a two-by-two matrix, as only the attack state is stochastic. At each time epoch t, the observation y of the core state x is obtained from the attack detection result. Since we assume the monitor adopts the Exp3.M-VP algorithm, given that the attacker may also adopt a learning-based algorithm to avoid being detected and according to Corollary 5.2, we estimate the information matrix $\mathbf{\Psi_a}$ under action a as follows,

$$\Psi_a(1|1) = \frac{a}{N} \cdot \gamma + \frac{N-a}{N} \cdot \beta \tag{6.5a}$$

$$\Psi_a(0|1) = \frac{N-a}{N} \cdot (1-\beta) + \frac{a}{N} \cdot (1-\gamma) \tag{6.5b}$$

$$\Psi_a(1|0) = \beta \tag{6.5c}$$

$$\Psi_a(0|0) = 1-\beta \tag{6.5d}$$

where γ represents detection recall/sensitivity, $(1-\gamma)$ represents false negative rate, β represents false positive rate, and $(1-\beta)$ represents true negative rate. Equation (6.5a) denotes the probability that the security monitor observes an attack, given the CAV is under attack. The returned positive detection can be either: (i) a *true positive* with probability γ if it scans the compromised sensor with probability a/N; or (ii) a *false positive* with probability β if it misses the compromised sensor with probability $(N-a)/N$. The probability a/N is estimated from the equilibrium state between the attacker and the monitor, where in the equilibrium state the best strategy for the attacker is to randomly choose each sensor with the same probability. Equation (6.5b) denotes the probability that the security monitor does not observe any attack, given the CAV is under attack. In this case, the

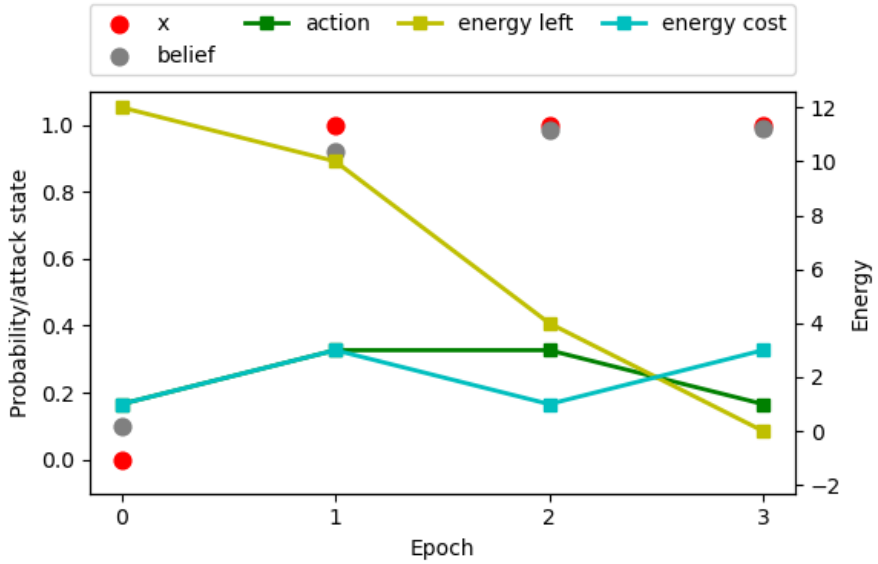


Figure 6.3: Sample path of the system states and the corresponding belief under PBVI policy.

returned negative detection can be either: (i) a *false negative* with probability $(1-\gamma)$ if it scans the compromised sensor with probability a/N; or (ii) a *true positive* with probability $(1-\beta)$ if it misses the compromised sensor with probability $(N-a)/N$. Equation (6.5c) denotes the probability that the security monitor observes an attack, given the CAV is not under attack. Lastly, equation (6.5d) denotes the probability that the security monitor does not observe an attack, given the CAV is not under attack. For this small-sized problem we set $\gamma = 0.95$ and $\beta = 0.05$.

At each time epoch, given a units of allocated security resources, the expected cost function of CAV is defined as

$$\bar{C}(1,a) = c_A \cdot \left(\frac{a}{N} \cdot (1-\gamma) + \frac{N-a}{N} \right) \tag{6.6a}$$

$$\bar{C}(0,a) = 0 \tag{6.6b}$$

where equation (6.6a) represents the expected cost the CAV incurs for being unable to detect an existing attack. In our simulations, we set $c_A = 20$, $c_U = 5$, and $c_R = 0.1$.

To solve the small-sized POMDP problem, we implement the point based value iteration (PBVI) algorithm (Pineau et al., 2003). PBVI is an approximated value iteration algorithm. It selects a small set of representative belief points and then tracks the value and its derivative for the representative points only. PBVI provides a (near-) optimal policy. We then compare the results with those under two heuristic policies. For the first heuristic policy (denoted as H1), we always allocate only 1 unit of security energy, and for the second heuristic policy (denoted as H2), we always allocate 3 units of security energy, i.e. the security monitor always scans all sensors at each time epoch.

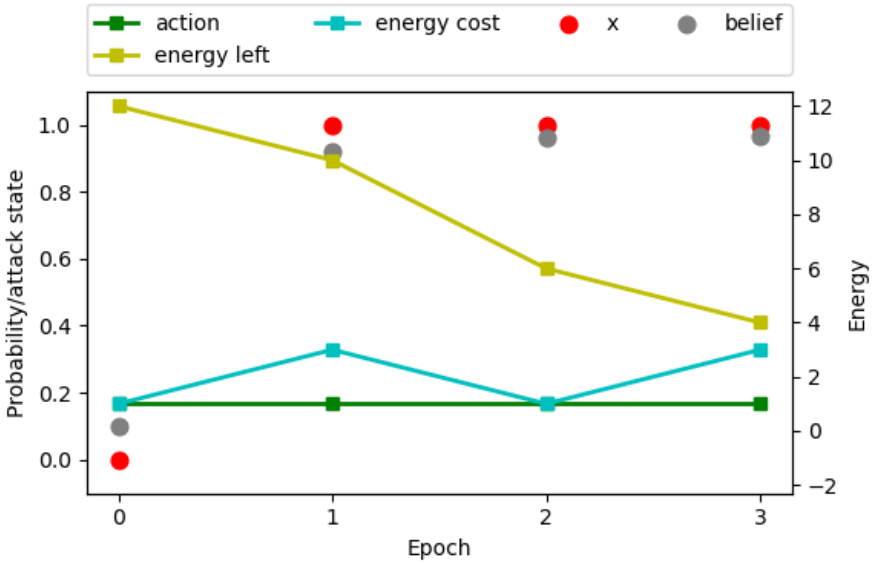


Figure 6.4: Sample path of the system states and the corresponding belief under H1 policy.

Figure 6.3 provides the sample paths under PBVI policy for the small example. This figure shows the attack state x, the belief of being attacked, the energy cost, and the remaining energy at each time epoch. We initialize the belief as $\pi_0(0) = 0.9$ and $\pi_0(1) = 0.1$. The attack occurs at epoch 1 and continues until the end of horizon. The belief of attack therefore jumps from $\pi_0(1) = 0.1$ at $t = 0$ to $\pi_1(1) = 0.9194$ at $t = 1$. Since higher number of security energy (i.e., a) increases both the observation accuracy and the cost function, and the fact that we set a relatively small reward (i.e., c_R) for the remaining energy at the end of the horizon compared with the penalties for the miss detection (i.e., c_A) and the unfinished trip (i.e., c_U), the PBVI policy uses the last unit of the remaining energy at the end of horizon to obtain a lower cost, which translates into strengthen its belief.

Figures 6.4 and 6.5 show the example sample paths under the H1 policy and the H2 policy, respectively. We see that under the H2 policy, the belief of the attack state is overall more accurate than the PBVI policy and the H1 policy, as we always allocate 3 units of security resources, which provides higher observation accuracy. However, under the H2 policy, the CAV runs out of the energy at $t = 3$ and receives an immediate penalty for the unfinished trip.

Figure 6.6 displays the cumulative cost under the PBVI, H1, and H2 policies. We can observe that PBVI achieves the lowest cost. Under the H1 policy, POMDP overall incurs the highest cost due to its inability to detect attacks, as the allocation of security resources is limited to only a single unit. Nonetheless, H1 receives a reward associated with more energy saving at the end of the horizon, evidenced by a reduction in cost at $t = 3$. However, since we set a relative small weights of c_R, it is not enough to compensate the overall cost caused by missed detection. The H2 policy achieve the

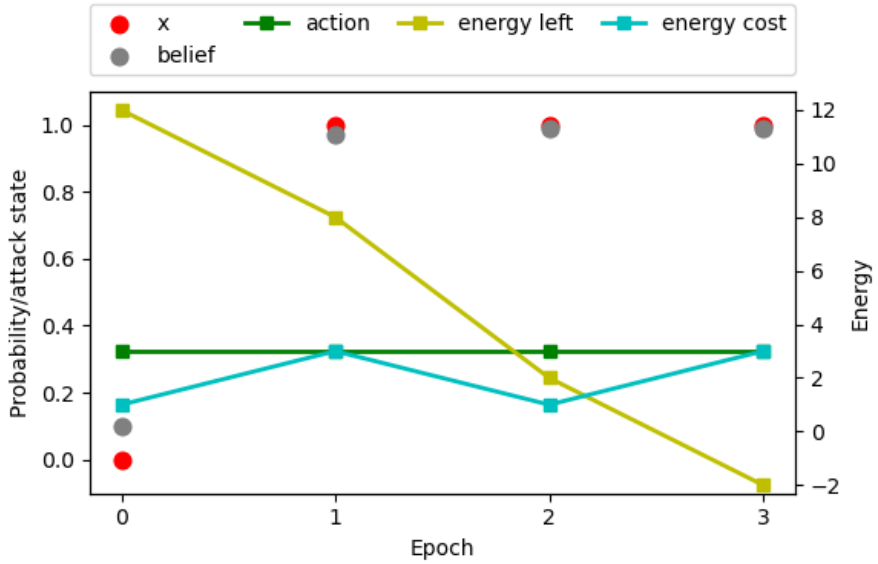


Figure 6.5: Sample path of the system states and the corresponding belief under H2 policy.

same level of cost as PBVI before $t = 3$. However, since the CAV runs out of the energy at $t = 3$, an immediate cost of c_U is added to the cost of the H2 policy.

6.5 Conclusion and Future Work

In this chapter, we develop a framework for dynamic security resource allocation for connected and automated vehicles. The work in this chapter can be a complementary supplement of Chapter 5, where the Exp3.M-VP algorithm can be used as the attack detection algorithm. We consider the fact that the security monitor may not be able to perfectly observe the attack state but only partially observes it using the attack detection algorithm. The goal of this chapter is to dynamically allocate security resources, in the form of energy units, to maintain the safety of the CAV while ensuring that the trip can be completed, based on the observed system state.

Specifically, we formulate a partially observable Markov decision process (POMDP) model to prescribe a security resource allocation policy that minimizes the total discounted cost. In a small example, we consider a CAV with 3 candidate sensors subject to potential attacks. We solve the problem to near-optimality using the PBVI algorithm. By comparing the obtained policy with two heuristic polices of scanning only 1 or all sensors in each time epoch, we demonstrate the efficacy of the proposed approach. Several potential directions can be identified for this work: First, a more diverse set of problems, including larger-sized problems, should be explored by considering more candidate sensors and a longer time horizon. Also it is worthwhile to consider a time-variant TPM, as in practice the state-transition probability can be time-dependent. Since PBVI is not

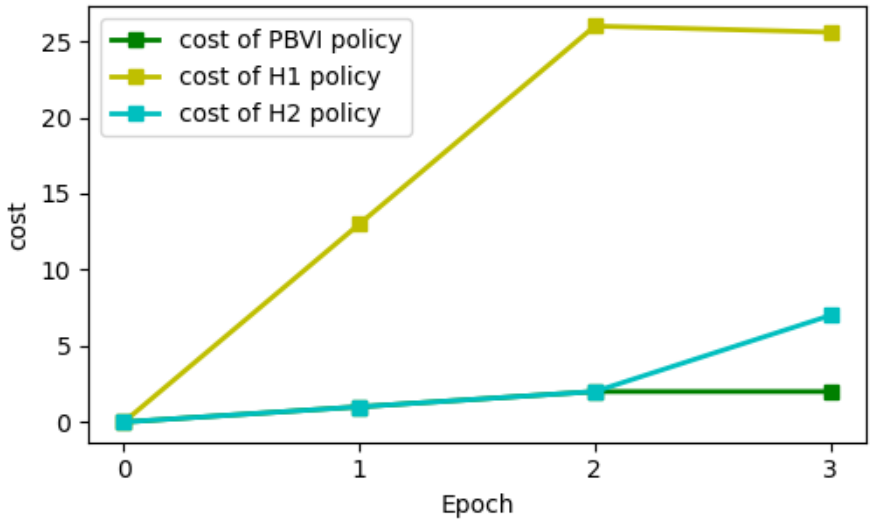


Figure 6.6: Cumulative cost under PBVI, H1, and H2.

suitable for solving large-scale POMDP models with high-dimensional action spaces, reinforcement learning algorithms such as asynchronous advantage actor critic (A3C) should be explored for solving large-scale POMDP models. Secondly, an extension may consider additional features for security resource allocation. Examples include modeling traffic conditions and location information. Lastly, it would be beneficial to use real-world data for a more accurate estimation of TPM and the information matrix.

CHAPTER 7

Conclusion and Discussions

7.1 Research Summary and Findings

Due the the recent advance in information and communication technologies (ICT) and autonomous driving research, CAVs are recognized as an integral part of the next generation of transportation systems, prompting many academic institutions, research centers, and the private sector to conduct research and contribute to this field in exceedingly high levels. Compared with traditional vehicles, CAVs are expected to cooperate by leveraging V2V communications between vehicles, and V2I/I2V connections between vehicles and infrastructures. As the cooperative exchange of data will provide vital inputs to improve the performance and safety of the automation systems, it becomes important to ensure the cybersecurity of the connected and automated transportation systems. This dissertation aims to address some of the most prominent challenges in CAV cybersecurity by developing solutions that leverage advancements in machine learning, reinforcement learning, and algorithmic game theory. The cybersecurity challenges faced by CAVs can be roughly divided into two major components.

7.1.1 Anomaly Detection in CAV Sensors

The first part of the dissertation addresses the set of challenges a CAV may face in collecting data used to perceive the environment. These challenges include potential sensor faults, malicious cyberattacks, and time delays introduced through the communication chanel and/or delays in sensor perception and/or input processing. In Chapters 2-4, we address these challenges by developing a model-based filtering framework to securely estimate a CAV's sensor state by utilizing surrounding vehicles' information in different scenarios. Specifically, Chapters 2 and 3 consider a car-following scenario and Chapter 4 consider a platooning scenario. Chapter 3 extends Chapter 2 by improving the filtering and detection performance and considering a stochastic time delay factor. More specifically, in the first part of the dissertation, we devise a filtering-detection framework by combining an extended Kalman filter (EKF)-based algorithm with a semi-supervised learning algorithm to smooth

sensor readings as well as detect sensor anomalies. Our experiments show that the proposed method can significantly improve the detection performance compared with classic detection methods.

As we consider a platooning scenario in Chapter 4, we further analyze platoon stability under cybersecurity uncertainties. We first analyze the head-to-tail stability with different attack parameters, which provides insights on how a platoon performs under various attack scenarios. We further propose a new class measures for string stability, which we call pseudo string stability. Pseudo string stability takes into account the detection recall/sensitivity, and allows for establishing a relationship between attack parameters and detection sensitivity. We obtain the critical detection sensitivities that guarantee pseudo string stability under various attack parameters. Such information could be used to configure the detector under different attack scenarios.

7.1.2 Security Monitoring in Connected and Automated Transportation Systems

The second part of this dissertation aims to address the challenges regarding the unknown behavior of the adversary and constrained security resource. Prevention is considered to be one of the best defense strategies against malicious hackers, and predicting a strategic adversary's attack profile is an important stepping stone to deploy better prevention mechanisms. Moreover, due to the limited computational ability, battery supply, and communication bandwidth at the disposal of a CAV, the security monitoring systems can be subject to constrained resources. As such, CAVs need to be equipped with a dynamic security resource allocation scheme to ensure both security and energy efficiency in a non-deterministic environment.

In Chapter 5, we devise a adversarial/non-stochastic multi-play multi-armed bandit (MPMAB) algorithm to predict the adversary's behavior under time-variant security resources. We provide a sublinear regret bound of the proposed algorithm. We model both the attacker and the defender (security monitor) in a game theoretical setting and derive the condition under which a Nash equilibrium of the strategic game exists. Chapter 5 validates two directions from a game theoretical perspective in order to improve security in connected and automated transportation systems: (i) allocating more security monitoring resource; or (ii) improving the robustness of cybersecurity solutions, e.g., increasing detection recall/sensitivity.

Chapter 6 address the last challenge, namely, constrained security resource. To address this challenge, we formulate a dynamic resource allocation problem to allocate security resources to a CAV in real-time based on the observed system state. Specifically, we consider an imperfect security monitor that provides partial observation of the system state, i.e., whether the CAV is under attack, and formulate the problem as a partially observable Markov decision process (POMDP) with the objective of minimizing the total discounted travel cost. Our experiments demonstrate the

effectiveness of the proposed model. This dynamic resource allocation problem can be considered as a complementary supplement to the resource-constrained bandit problem in Chapter 5.

7.2 Directions for Future Work

At the end of Chapters 2-6, we provide the future directions to further extend the methodologies/experiments presented therein. In what follows, we provide additional topics related to the content of this dissertation that are worth exploring in further detail.

Real-World Implementation of Detection Algorithms: In Chapters 2-4, we consider anomaly detection in CAV sensors under different scenarios, where we only consider longitudinal models to represent vehicles' dynamics. The longitudinal dynamics may be suitable for theoretical analysis of vehicle strings, but it cannot fully represent the complex nature of a vehicle's motion in practice. In order to implement the proposed framework in the real world, we need to adopt motion models that more accurately represent actual vehicle dynamics.

Anomaly Detection Using Multi-Modal Data: Throughout Chapters 2-4, we focus on sensor anomaly detection in CAVs, where we assume data from different sensors are synchronized, collected at the same rate, and converted to time-series data. However, in practice, what we receive can be multi-model data including images from cameras, point cloud data from LiDAR systems, and time series from other onboard sensors, where various sources of data can be asynchronous. Current literature in the field of autonomous driving has considered multi-modal sensor fusion for end-to-end autonomous driving (Huang et al., 2020; Xiao et al., 2020; Prakash et al., 2021) and object detection (Feng et al., 2020; Chiu et al., 2021). However, there is still a paucity of anomaly detection/cybersecurity research leveraging multi-modal data. Exploration of multi-modal data collected from CAVs and other sources (e.g., RSUs) for anomaly detection and signal recovery could make cybersecurity solutions more practical for real world implementation.

Holistic Framework for Dynamic Security Resource Allocation and Sensor Selection: In Chapters 5-6, we consider macroscopic decision-making problems that focus on cybersecurity of connected and automated transportation systems, where we divide the dynamic sensor selection and dynamic security resource allocation into two different tasks, i.e., MPMAB and POMDP, respectively. One advantage of using MPMAB is that it provides theoretical guarantee of asymptotic performance. Meanwhile, POMDP provides us the flexibility to consider imperfect observations. Although there exist numerous reinforcement learning based algorithms for solving large-scale POMDP models, in the context of our problem, it is hard to obtain the real-world data for training, and to obtain a theoretical performance guarantee of the learned policy. It would be worthwhile to explore online

learning approaches, which are able to provide a theoretical guarantee on asymptotic performance, to create a holistic framework for both dynamic security resource allocation and sensor selection. For example, the contextual bandit (Li et al., 2010) is a variant of the multi-armed bandit problem that associates side information/context with each arm. Based on this side information, which is in the form of features, the algorithm needs to learn a policy that maps the feature space to arms. It is of interest to explore whether we can regard the states of the dynamic security resource allocation task as the side information, and devise a variable-played version of the contextual bandit.

APPENDIX A

Supporting Materials of Chapter 5

A.1 Hedge and Exp3 Algorithms

Algorithm A.1: **Hedge**

1: Parameters: $\iota \in \mathbb{R}^+$.
2: Initialization: Set $r_k(1) := 0$ for all $k \in \mathcal{N}$.
3: **for** t = 1,2,...,T **do**
4: Choose action I_t according to the distribution

$$\beta_k = \frac{(1+\iota)^{r_k(t)}}{\sum_{j=1}^{N}(1+\iota)^{r_j(t)}}.$$

5: Receive the reward vector $x(t)$ and score gain $x_{I_t}(t)$.
6: Set $r_k(t+1) := r_k(t) + x_k(t)$ for all $k \in \mathcal{N}$.
7: **end for**

Algorithm A.2: **Exp3**

1: Parameters: $\iota \in \mathbb{R}^+$ and $\eta \in [0, 1]$.

2: Initialization: Initialize **Hedge**.

3: **for** t = 1,2,...,T **do**

4: Obtain the distribution vector $\beta(t) = (\beta_k(t), k \in \mathcal{N})$ from **Hedge**.

5: Select action I_t to be k with probability

$$\hat{\beta}_k(t) = (1 - \eta)\beta_k(t) + \eta/N.$$

6: Receive the reward $x_{I_t}(t) \in [0, 1]$.

7: Return the simulated reward vector $\hat{x}(t) = (\hat{x}_k(t), k \in \mathcal{N})$ to **Hedge** with

$$\hat{x}_k(t) = \begin{cases} \frac{\eta}{N} \times \frac{x_{I_t}(t)}{\hat{\beta}_{I_t}(t)} & \text{if } k = I_t \\ 0 & \text{otherwise.} \end{cases}$$

8: **end for**

A.2 DepRound Algorithm

Algorithm A.3: **DepRound:** The Dependent Rounding Algorithm

Inputs: Natural number $M < N$, marginal distribution $(p_k, k \in \mathcal{N})$ with $\sum_{k=1}^{N} p_k = M$

Output: Subset $\mathcal{N}_1$ of $\mathcal{N}$ such that $|\mathcal{N}_1| = M$

while $\{k \in \mathcal{N} : 0 < p_k < 1\} \neq \emptyset$ **do**

  Choose distinct i and j such that $0 < p_i < 1$ and $0 < p_j < 1$

  Set $\rho = \min\{1 - p_i, p_j\}$ and $\zeta = \min\{p_i, 1 - p_j\}$

  Update p_i and p_j as

$$(p_i, p_j) = \begin{cases} (p_i + \rho, p_j - \rho) & \text{with probability } \frac{\zeta}{\rho+\zeta} \\ (p_i - \zeta, p_j + \zeta) & \text{with probability } \frac{\rho}{\rho+\zeta} \end{cases}$$

end while

return $\{k : p_k = 1, 1 \leq k \leq N\}$

A.3 Proof of Theorem 5.2

Proof. Note that

$$\begin{aligned}
\bar{r}_\infty =& 1 - \liminf_{T\to\infty} \mathbb{E}\left[\frac{1}{T} G^J_{\text{Exp3.M-VP}}(T)\right] && \text{(A.1a)}\\
\leq& 1 - \liminf_{T\to\infty} \frac{1}{T}\left(G_{\max}(T) - 2\sqrt{\left(1+(e-2)\frac{b}{a}\right)}\sqrt{bTN\ln\frac{N}{b}}\right) && \text{(A.1b)}\\
=& 1 - \liminf_{T\to\infty} \frac{1}{T} G_{\max}(T) && \text{(A.1c)}\\
\leq& \frac{N-a}{N} && \text{(A.1d)}
\end{aligned}$$

for any policy π of the attacker, where the last inequality (A.1d) comes from the fact that $G_{\max}(T) \geq \frac{Ta}{N}$ for any defender's policy γ.

Under the greedy policy we have $\hat{\beta}_{g(t)}(t) \leq \frac{b}{N}$, which implies $r(t) \geq \frac{N-b}{N}$ for any t. Therefore by using the greedy policy π_{greedy}, we have $\bar{r}_\infty \geq \frac{N-b}{N}$. □

A.4 Proof of Lemma 5.2

The proof of Lemma 5.2 is an extension of the proof of Lemma 4 in (Wang & Liu, 2015). The main difference is that the matrix $\mathbf{H}$ is now an $N \times |\mathcal{C}(\mathcal{N}, a)|$ matrix compared to the one in the original proof which is $N \times N$.

Proof. 1) The problem (5.27) is equivalent to

$$\max_{d \in \Delta_N} \min_{u \in \Delta_N} \sum_{n=1}^{|\mathcal{C}(\mathcal{N},a)|} \sum_{k=1}^{N} \left(\mu_k d_k - \sum_{j \in J_n} \mu_j d_j \right) u_n \tag{A.2}$$

which can be rewritten in matrix form:

$$\max_{\mathbf{d} \in \Delta_N} \min_{\mathbf{u} \in \Delta_N} \mathbf{d}^\intercal \mathbf{H} \mathbf{u} \tag{A.3}$$

where $\intercal$ denotes the transpose, and

$$\mathbf{H} = \begin{bmatrix} 0 & \mu_1 & \cdots & \mu_1 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mu_a \\ \mu_{a+1} & 0 & \cdots & \mu_{a+1} \\ \vdots & \vdots & \ddots & \vdots \\ \mu_{N-a+1} & \mu_{N-a+1} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \mu_N & \mu_N & \cdots & 0 \end{bmatrix}$$

where each column j represents one set $\mathcal{S} \subseteq \mathcal{C}(\mathcal{N}, a)$ such that for all $i \in \mathcal{S}$, $H_{ij} = 0$ and for all $i \in \mathcal{N} \setminus \mathcal{S}$, $H_{ij} = \mu_i$. The remaining proof is the same as the original proof.

Now consider a zero-sum game with the payoff matrices for the row and the column players being $\mathbf{H}$ and $-\mathbf{H}$, whose mixed strategy vectors are $\mathbf{d}$ and $\mathbf{u}$, respectively. Any optimal solution $\mathbf{d}^*$ to the problem (5.27) is a Nash equilibrium strategy for the row player, and by the indifference condition, we obtain for any $j \in \text{supp}(\mathbf{d}^*)$,

$$\sum_{k \neq j} k \in \text{supp}(\mathbf{d}^*) = \text{Const.} \tag{A.4}$$

which implies $\mu_k d_k^* = \mu_j d_j^*$ for any $k, j \in \text{supp}(\mathbf{d}^*)$.

2) The second part of the Lemma is proved by contradiction. Assume that there exist $i \in \text{supp}(\mathbf{d}^*)$ and $j \in \mathcal{N} \setminus \text{supp}(\mathbf{d}^*)$ such that $\mu_j > \mu_i$. Let o be a constant such that

$o = \mu_k d_k^*$ for any $k \in \text{supp}(\mathbf{d}^*)$. Then consider a feasible solution $\mathbf{d}$, where $d_k = 0$ for all $k \in ((\mathcal{N} \setminus \text{supp}(\mathbf{d}^*)) \setminus \{j\}) \cup \{i\}$, and $d_k = d_k^* + \epsilon$ for all $k \in (\text{supp}(\mathbf{d}^*) \setminus \{i\}) \cup \{j\}$, with $\epsilon = d_i^*(1 - \mu_i/\mu_j)/K^*$, which yields a higher objective value. □

A.5 Proof of Lemma 5.3

Proof. Consider the following linear program:

$$\underset{\mathbf{l},p}{\text{minimize}}\ p \tag{A.5a}$$

$$\text{s.t. } p + \mu_k l_k \leq \mu_k \tag{A.5b}$$

$$\sum_k l_k \leq b \tag{A.5c}$$

$$0 \leq l_k \leq 1 \tag{A.5d}$$

It is easy to see that problem (5.32) is lower bounded by the problem (A.5).

Then the dual of the program (A.5) can be written as

$$\underset{\mathbf{d},q}{\text{maximize}}\ \sum_{k=1}^{N} \mu_k d_k - q \tag{A.6a}$$

$$\text{s.t. } \sum_{k=1}^{N} \mu_k d_k \leq \frac{Nq}{b} \tag{A.6b}$$

$$\sum_{k=1}^{N} d_k = 1 \tag{A.6c}$$

$$d_k \geq 0 \tag{A.6d}$$

Note that program (A.6) is equivalent to the following problem

$$\underset{\mathbf{d}\in\Delta_N}{\text{maximize}}\ \sum_{k=1}^{N} \mu_k d_k - \max_{J\in\mathcal{C}(\mathcal{N},b)} \sum_{j\in J} \mu_j d_j \tag{A.7}$$

which is essentially problem (5.27), except for changing the set $\mathcal{C}(\mathcal{N}, a)$ to $\mathcal{C}(\mathcal{N}, b)$. Therefore problem (A.7) has the optimal value $\frac{K^*-b}{\sum_{k=1}^{K^*}\mu_k}$, which provides us with the lower bound. □

BIBLIOGRAPHY

Abdolmaleki, M., Masoud, N., & Yin, Y. (2019). Vehicle-to-vehicle wireless power transfer: Paving the way toward an electrified transportation system. *Transportation Research Part C: Emerging Technologies*, 103, 261–280.

Abdolmaleki, M., Shahabi, M., Yin, Y., & Masoud, N. (2021). Itinerary planning for cooperative truck platooning. *Transportation Research Part B: Methodological*, 153, 91–110.

Abernethy, J. D., Hazan, E., & Rakhlin, A. (2008). Competing in the dark: An efficient algorithm for bandit linear optimization. *21st Annual Conference on Learning Theory - COLT 2008, Helsinki, Finland, July 9-12, 2008*, 263–274.

Agrawal, R., Hegde, M., & Teneketzis, D. (1990). Multi-armed bandit problems with multiple plays and switching cost. *Stochastics and Stochastic reports*, 29(4), 437–459.

Akhlaghi, S., Zhou, N., & Huang, Z. (2017). Adaptive adjustment of noise covariance in kalman filter for dynamic state estimation. *Power & Energy Society General Meeting, 2017 IEEE*, 1–5.

Alam, A. (2011). *Fuel-efficient distributed control for heavy duty vehicle platooning*. KTH Royal Institute of Technology.

Alipour-Fanid, A., Dabaghchian, M., Zhang, H., & Zeng, K. (2017). String stability analysis of cooperative adaptive cruise control under jamming attacks. *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*, 157–162.

Alotibi, F. & Abdelhakim, M. (2020). Anomaly detection for cooperative adaptive cruise control in autonomous vehicles using statistical learning and kinematic model. *IEEE Transactions on Intelligent Transportation Systems*, 22(6), 3468–3478.

Alpcan, T. & Buchegger, S. (2010). Security games for vehicular networks. *IEEE Transactions on Mobile Computing*, 10(2), 280–290.

Anantharam, V., Varaiya, P., & Walrand, J. (1987). Asymptotically efficient allocation rules for the multiarmed bandit problem with multiple plays-part i: Iid rewards. *IEEE Transactions on Automatic Control*, 32(11), 968–976.

Arroyo-Valles, R., Marques, A. G., & Cid-Sueiro, J. (2007). Energy-aware geographic forwarding of prioritized messages in wireless sensor networks. *2007 IEEE International Conference on Mobile Adhoc and Sensor Systems*, 1–9. https://doi.org/10.1109/MOBHOC.2007.4428652

Auer, P., Cesa-Bianchi, N., & Fischer, P. (2002a). Finite-time analysis of the multiarmed bandit problem. *Machine learning*, 47(2-3), 235–256.

Auer, P., Cesa-Bianchi, N., Freund, Y., & Schapire, R. E. (1995). Gambling in a rigged casino: The adversarial multi-armed bandit problem. *Proceedings of IEEE 36th annual foundations of computer science*, 322–331.

Auer, P., Cesa-Bianchi, N., Freund, Y., & Schapire, R. E. (2002b). The nonstochastic multiarmed bandit problem. *SIAM journal on computing*, 32(1), 48–77.

Bahamou, S., El Ouadghiri, M. D., & Bonnin, J.-M. (2016). When game theory meets vanet's security and privacy. *proceedings of the 14th international conference on advances in mobile computing and multi media*, 292–297.

Bar-Shalom, Y. & Li, X.-R. (1995). *Multitarget-multisensor tracking: principles and techniques*, volume 19. YBs Storrs, CT.

Baskiotis, C., Raymond, J., & Rault, A. (1979). Parameter identification and discriminant analysis for jet engine machanical state diagnosis. *Decision and Control including the Symposium on Adaptive Processes, 1979 18th IEEE Conference on*, volume 2, 648–650.

Bezzina, D. & Sayer, J. (2014). Safety pilot model deployment: Test conductor team report. *Report No. DOT HS*, 812, 171.

Biron, Z. A., Dey, S., & Pisu, P. (2018). Real-time detection and estimation of denial of service attack in connected vehicle systems. *IEEE Transactions on Intelligent Transportation Systems*, 19(12), 3893–3902.

Biroon, R. A., Biron, Z. A., & Pisu, P. (2021). False data injection attack in a platoon of cacc: real-time detection and isolation with a pde approach. *IEEE Transactions on Intelligent Transportation Systems*.

Brahmi, I. H., Ansari, N., Rehmani, M. H., et al. (2019). Cyber security framework for vehicular network based on a hierarchical game. *IEEE Transactions on Emerging Topics in Computing*.

Brumback, B. & Srinath, M. (1987). A chi-square test for fault-detection in kalman filters. *IEEE Transactions on Automatic Control*, 32(6), 552–554.

Bubeck, S., Cesa-Bianchi, N., et al. (2012). Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends® in Machine Learning*, 5(1), 1–122.

Cao, Y., Xiao, C., Cyr, B., Zhou, Y., Park, W., Rampazzi, S., Chen, Q. A., Fu, K., & Mao, Z. M. (2019a). Adversarial sensor attack on lidar-based perception in autonomous driving. *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*, 2267–2281.

Cao, Y., Xiao, C., Yang, D., Fang, J., Yang, R., Liu, M., & Li, B. (2019b). Adversarial objects against lidar-based autonomous driving systems. *arXiv preprint arXiv:1907.05418*.

Cesa-Bianchi, N. & Lugosi, G. (2012). Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5), 1404–1422.

Chandra, A., Gong, W., & Shenoy, P. (2003). Dynamic resource allocation for shared data centers using online measurements. *Quality of Service — IWQoS 2003*, 381–398.

Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., Savage, S., Koscher, K., Czeskis, A., Roesner, F., Kohno, T., et al. (2011). Comprehensive experimental analyses of automotive attack surfaces. *USENIX Security Symposium*, 77–92.

Cheng, M., Ruan, L., & Wu, W. (2005). Achieving minimum coverage breach under bandwidth constraints in wireless sensor networks. *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, volume 4, 2638–2645 vol. 4. https://doi.org/10.1109/INFCOM.2005.1498547

Chiu, H.-k., Li, J., Ambruş, R., & Bohg, J. (2021). Probabilistic 3d multi-modal, multi-object tracking for autonomous driving. *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 14227–14233.

Christiansen, P., Nielsen, L. N., Steen, K. A., Jørgensen, R. N., & Karstoft, H. (2016). Deepanomaly: Combining background subtraction and deep learning for detecting obstacles and anomalies in an agricultural field. *Sensors*, 16(11), 1904.

Clark, R., Fosth, D. C., & Walton, V. M. (1975). Detecting instrument malfunctions in control systems. *IEEE Transactions on Aerospace and Electronic Systems*, AES-11(4), 465–473. https://doi.org/10.1109/TAES.1975.308108

Cui, L., Hu, J., Park, B. B., & Bujanovic, P. (2018). Development of a simulation platform for safety impact analysis considering vehicle dynamics, sensor errors, and communication latencies: Assessing cooperative adaptive cruise control under cyber attack. *Transportation research part C: emerging technologies*, 97, 1–22.

Deckert, J., Desai, M., Deyst, J., & Willsky, A. (1977). F-8 dfbw sensor failure identification using analytic redundancy. *IEEE Transactions on Automatic Control*, 22(5), 795–803.

Faughnan, M. S., Hourican, B. J., MacDonald, G. C., Srivastava, M., Wright, J.-P. A., Haimes, Y. Y., Andrijcic, E., Guo, Z., & White, J. C. (2013). Risk analysis of unmanned aerial vehicle hijacking and methods of its detection. *Systems and Information Engineering Design Symposium (SIEDS), 2013 IEEE*, 145–150.

Feng, D., Haase-Schütz, C., Rosenbaum, L., Hertlein, H., Glaeser, C., Timm, F., Wiesbeck, W., & Dietmayer, K. (2020). Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, 22(3), 1341–1360.

Feng, S., Zhang, Y., Li, S. E., Cao, Z., Liu, H. X., & Li, L. (2019). String stability for vehicular platoon control: Definitions and analysis methods. *Annual Reviews in Control*, 47, 81–97.

Feng, Y., Huang, S., Chen, Q. A., Liu, H. X., & Mao, Z. M. (2018). Vulnerability of traffic control system under cyberattacks with falsified data. *Transportation research record*, 2672(1), 1–11.

Fok, E. (2013). An introduction to cybersecurity issues in modern transportation systems. *ITE Journal*, 3, 19.

Fouché, E., Komiyama, J., & Böhm, K. (2019). Scaling multi-armed bandit algorithms. *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 1449–1459.

Freund, Y. & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, 55(1), 119–139. https://doi.org/https://doi.org/10.1006/jcss.1997.1504

Gandhi, R., Khuller, S., Parthasarathy, S., & Srinivasan, A. (2006). Dependent rounding and its applications to approximation algorithms. *Journal of the ACM (JACM)*, 53(3), 324–360.

Geng, Y. & Wang, J. (2008). Adaptive estimation of multiple fading factors in kalman filter for navigation applications. *Gps Solutions*, 12(4), 273–279.

Gertler, J. (1997). Fault detection and isolation using parity relations. *Control engineering practice*, 5(5), 653–661.

Guegan, L. & Orgerie, A.-C. (2019). Estimating the end-to-end energy consumption of low-bandwidth iot applications for wifi devices. *2019 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, 287–294. https://doi.org/10.1109/CloudCom.2019.00049

Gunduz, D. & Erkip, E. (2007). Opportunistic cooperation by dynamic resource allocation. *IEEE Transactions on Wireless Communications*, 6(4), 1446–1454. https://doi.org/10.1109/TWC.2007.348341

Guo, G. & Wen, S. (2015). Communication scheduling and control of a platoon of vehicles in vanets. *IEEE Transactions on intelligent transportation systems*, 17(6), 1551–1563.

György, A., Linder, T., Lugosi, G., & Ottucsák, G. (2007). The on-line shortest path problem under partial monitoring. *Journal of Machine Learning Research*, 8(Oct), 2369–2403.

Hill, D. J. & Minsker, B. S. (2010). Anomaly detection in streaming environmental sensor data: A data-driven modeling approach. *Environmental Modelling & Software*, 25(9), 1014–1022.

Hill, D. J., Minsker, B. S., & Amir, E. (2007). Real-time bayesian anomaly detection for environmental sensor data. *Proceedings of the Congress-International Association for Hydraulic Research*, volume 32, 503.

Huang, Z., Lv, C., Xing, Y., & Wu, J. (2020). Multi-modal sensor fusion-based deep neural network for end-to-end autonomous driving with scene understanding. *IEEE Sensors Journal*, 21(10), 11781–11790.

Humpherys, J., Redd, P., & West, J. (2012). A fresh look at the kalman filter. *SIAM review*, 54(4), 801–823.

Hwang, I., Kim, S., Kim, Y., & Seah, C. E. (2010). A survey of fault detection, isolation, and reconfiguration methods. *IEEE transactions on control systems technology*, 18(3), 636–653.

Ibanez, J. G., Zeadally, S., & Contreras-Castillo, J. (2015). Integration challenges of intelligent transportation systems with connected vehicle, cloud computing, and internet of things technologies. *IEEE Wireless Communications*, 6(22), 122–128.

Isermann, R. (1984). Process fault detection based on modeling and estimation methods—a survey. *automatica*, 20(4), 387–404.

Jin, I. G. & Orosz, G. (2014). Dynamics of connected vehicle systems with delayed acceleration feedback. *Transportation Research Part C: Emerging Technologies*, 46, 46–64.

Ju, Z., Zhang, H., & Tan, Y. (2020). Distributed deception attack detection in platoon-based connected vehicle systems. *IEEE transactions on vehicular technology*, 69(5), 4609–4620.

Kelarestaghi, K. B., Heaslip, K., Khalilikhah, M., Fuentes, A., & Fessmann, V. (2018). Intelligent transportation system security: hacked message signs. *SAE International Journal of Transportation Cybersecurity and Privacy*, 1(11-01-02-0004), 75–90.

Khattak, Z. H., Smith, B. L., & Fontaine, M. D. (2021). Impact of cyberattacks on safety and stability of connected and automated vehicle platoons under lane changes. *Accident Analysis & Prevention*, 150, 105861.

Kiekintveld, C., Jain, M., Tsai, J., Pita, J., Ordóñez, F., & Tambe, M. (2009). Computing optimal randomized resource allocations for massive security games. *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 689–696.

Kim, W. & Jung, I. (2019). Smart sensing period for efficient energy consumption in iot network. *Sensors*, 19(22). https://doi.org/10.3390/s19224915

Kocić, J., Jovičić, N., & Drndarević, V. (2018). Sensors and sensor fusion in autonomous vehicles. *2018 26th Telecommunications Forum (TELFOR)*, 420–425. https://doi.org/10.1109/TELFOR.2018.8612054

Komiyama, J., Honda, J., & Nakagawa, H. (2015). Optimal regret analysis of thompson sampling in stochastic multi-armed bandit problem with multiple plays. *arXiv preprint arXiv:1506.00779*.

Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., et al. (2010). Experimental security analysis of a modern automobile. *Security and Privacy (SP), 2010 IEEE Symposium on*, 447–462.

Lesage-Landry, A. & Taylor, J. A. (2017). The multi-armed bandit with stochastic plays. *IEEE Transactions on Automatic Control*, 63(7), 2280–2286.

Li, L., Chu, W., Langford, J., & Schapire, R. E. (2010). A contextual-bandit approach to personalized news article recommendation. *Proceedings of the 19th International Conference on World Wide Web*, WWW '10, 661–670. https://doi.org/10.1145/1772690.1772758

Litman, T. (2017). *Autonomous vehicle implementation predictions*. Victoria Transport Policy Institute Victoria, Canada.

Liu, X., Masoud, N., Zhu, Q., & Khojandi, A. (2022). A markov decision process framework to incorporate network-level data in motion planning for connected and automated vehicles. *Transportation Research Part C: Emerging Technologies*, 136, 103550.

Liu, X., Zhao, G., Masoud, N., & Zhu, Q. (2021). Trajectory planning for connected and automated vehicles: Cruising, lane changing, and platooning. *SAE International Journal of Connected and Automated Vehicles*, 4(12-04-04-0025), 315–333.

Marchetti, M., Stabili, D., Guido, A., & Colajanni, M. (2016). Evaluation of anomaly detection for in-vehicle networks through information-theoretic algorithms. *2016 IEEE 2nd International Forum on Research and Technologies for Society and Industry Leveraging a better tomorrow (RTSI)*, 1–6.

Masoud, N. & Jayakrishnan, R. (2017). Autonomous or driver-less vehicles: Implementation strategies and operational concerns. *Transportation research part E: logistics and transportation review*, 108, 179–194.

Mejri, M. N., Achir, N., & Hamdi, M. (2016). A new security games based reaction algorithm against dos attacks in vanets. *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, 837–840.

Meyer, G. & Beiker, S. (2019). *Road vehicle automation 5*, volume 201955. Springer.

Mohamed, A. & Schwarz, K. (1999). Adaptive kalman filtering for ins/gps. *Journal of geodesy*, 73(4), 193–203.

Molnár, T. G., Qin, W. B., Insperger, T., & Orosz, G. (2015). Predictor design for connected cruise control subject to packet loss. *IFAC-PapersOnLine*, 48(12), 428–433.

Molnár, T. G., Qin, W. B., Insperger, T., & Orosz, G. (2017). Application of predictor feedback to compensate time delays in connected cruise control. *IEEE Transactions on Intelligent Transportation Systems*, 19(2), 545–559.

Mousavinejad, E., Yang, F., Han, Q.-L., Ge, X., & Vlacic, L. (2019). Distributed cyber attacks detection and recovery mechanism for vehicle platooning. *IEEE Transactions on Intelligent Transportation Systems*, 21(9), 3821–3834.

Mousavinejad, E., Yang, F., Han, Q.-L., Qiu, Q., & Vlacic, L. (2018). Cyber attack detection in platoon-based vehicular networked control systems. *2018 IEEE 27th International Symposium on Industrial Electronics (ISIE)*, 603–608.

Müter, M. & Asaj, N. (2011). Entropy-based anomaly detection for in-vehicle networks. *Intelligent Vehicles Symposium (IV), 2011 IEEE*, 1110–1115.

Müter, M., Groll, A., & Freiling, F. C. (2010). A structured approach to anomaly detection for in-vehicle networks. *Information Assurance and Security (IAS), 2010 Sixth International Conference on*, 92–98.

Nabar, S., Walling, J., & Poovendran, R. (2010). Minimizing energy consumption in body sensor networks via convex optimization. *2010 International Conference on Body Sensor Networks*, 62–67. https://doi.org/10.1109/BSN.2010.19

Navda, V., Bohra, A., Ganguly, S., & Rubenstein, D. (2007). Using channel hopping to increase 802.11 resilience to jamming attacks. *IEEE INFOCOM 2007-26th IEEE International Conference on Computer Communications*, 2526–2530.

Nayak, B. P., Hota, L., Kumar, A., Turuk, A. K., & Chong, P. H. J. (2022). Autonomous vehicles: Resource allocation, security, and data privacy. *IEEE Transactions on Green Communications and Networking*, 6(1), 117–131. https://doi.org/10.1109/TGCN.2021.3110822

Ngoduy, D. (2015). Linear stability of a generalized multi-anticipative car following model with time delays. *Communications in Nonlinear Science and Numerical Simulation*, 22(1-3), 420–426.

Ni, K., Ramanathan, N., Chehade, M. N. H., Balzano, L., Nair, S., Zahedi, S., Kohler, E., Pottie, G., Hansen, M., & Srivastava, M. (2009). Sensor network data fault types. *ACM Transactions on Sensor Networks (TOSN)*, 5(3), 1–29.

Nisan, N., Roughgarden, T., Tardos, E., & Vazirani, V. V. (2007). *Algorithmic Game Theory*. Cambridge University Press (Chapter 18, 461–486).

Njilla, L. L., Kamhoua, C. A., Kwiat, K. A., Hurley, P., & Pissinou, N. (2017). Cyber security resource allocation: A markov decision process approach. *2017 IEEE 18th International Symposium on High Assurance Systems Engineering (HASE)*, 49–52. https://doi.org/10.1109/HASE.2017.30

Nowakowski, C., O'Connell, J., Shladover, S. E., & Cody, D. (2010). Cooperative adaptive cruise control: Driver acceptance of following gap settings less than one second. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, 54(24), 2033–2037. https://doi.org/10.1177/154193121005402403

Oddi, G., Panfili, M., Pietrabissa, A., Zuccaro, L., & Suraci, V. (2013). A resource allocation algorithm of multi-cloud resources based on markov decision process. *2013 IEEE 5th International Conference on Cloud Computing Technology and Science*, volume 1, 130–135. https://doi.org/10.1109/CloudCom.2013.24

Park, J., Ivanov, R., Weimer, J., Pajic, M., & Lee, I. (2015). Sensor attack detection in the presence of transient faults. *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*, 1–10.

Parkinson, S., Ward, P., Wilson, K., & Miller, J. (2017). Cyber threats facing autonomous and connected vehicles: Future challenges. *IEEE transactions on intelligent transportation systems*, 18(11), 2898–2915.

Petit, J. & Shladover, S. E. (2014). Potential cyberattacks on automated vehicles. *IEEE Transactions on Intelligent transportation systems*, 16(2), 546–556.

Petit, J. & Shladover, S. E. (2015). Potential cyberattacks on automated vehicles. *IEEE Trans. Intelligent Transportation Systems*, 16(2), 546–556.

Pineau, J., Gordon, G., Thrun, S., et al. (2003). Point-based value iteration: An anytime algorithm for pomdps. *Ijcai*, volume 3, 1025–1032.

Ploeg, J., Scheepers, B. T., Van Nunen, E., Van de Wouw, N., & Nijmeijer, H. (2011). Design and experimental evaluation of cooperative adaptive cruise control. *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, 260–265.

Pous, N., Gingras, D., & Gruyer, D. (2017). Intelligent vehicle embedded sensors fault detection and isolation using analytical redundancy and nonlinear transformations. *Journal of Control Science and Engineering*, 2017.

Prakash, A., Chitta, K., & Geiger, A. (2021). Multi-modal fusion transformer for end-to-end autonomous driving. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7077–7087.

Rajasegarar, S., Leckie, C., & Palaniswami, M. (2008). Anomaly detection in wireless sensor networks. *IEEE Wireless Communications*, 15(4).

Ran, B. & Boyce, D. (2012). *Dynamic urban transportation network models: theory and implications for intelligent vehicle-highway systems*, volume 417. Springer Science & Business Media.

Realpe, M., Vintimilla, B., & Vlacic, L. (2015). Sensor fault detection and diagnosis for autonomous vehicles. *MATEC Web of Conferences*, volume 30, 04003.

Ribeiro, M. I. (2004). Kalman and extended kalman filters: Concept, derivation and properties. *Institute for Systems and Robotics*, 43, 46.

Saghafian, S., Tomlin, B., & Biller, S. (2022). The internet of things and information fusion: who talks to who? *Manufacturing & Service Operations Management*, 24(1), 333–351.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., & Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7), 1443–1471.

Sedjelmaci, H., Hadji, M., & Ansari, N. (2019). Cyber security game for intelligent transportation systems. *IEEE Network*, 33(4), 216–222.

Sedjelmaci, H., Senouci, S. M., & Ansari, N. (2016). Intrusion detection and ejection framework against lethal attacks in uav-aided networks: A bayesian game-theoretic methodology. *IEEE Transactions on Intelligent Transportation Systems*, 18(5), 1143–1153.

Seo, E., Song, H. M., & Kim, H. K. (2018). Gids: Gan based intrusion detection system for in-vehicle network. *2018 16th Annual Conference on Privacy, Security and Trust (PST)*, 1–6. https://doi.org/10.1109/PST.2018.8514157

Sharma, A. B., Golubchik, L., & Govindan, R. (2010). Sensor faults: Detection methods and prevalence in real-world datasets. *ACM Transactions on Sensor Networks (TOSN)*, 6(3), 1–39.

Shida, M. & Nemoto, Y. (2009). Development of a small-distance vehicle platooning system. *16th ITS World Congress and Exhibition on Intelligent Transport Systems and ServicesITS AmericaERTICOITS Japan*.

Shladover, S. E. (2018). Connected and automated vehicle systems: Introduction and overview. *Journal of Intelligent Transportation Systems*, 22(3), 190–200.

Singh, A. K., Dziurzanski, P., Mendis, H. R., & Indrusiak, L. S. (2017). A survey and comparative study of hard and soft real-time dynamic resource allocation strategies for multi-/many-core systems. *ACM Comput. Surv.*, 50(2). https://doi.org/10.1145/3057267

Sinha, A., Nguyen, T. H., Kar, D., Brown, M., Tambe, M., & Jiang, A. X. (2015). From physical security to cybersecurity. *Journal of Cybersecurity*, 1(1), 19–35.

Song, N.-O. & Teneketzis, D. (2004). Discrete search with multiple sensors. *Mathematical Methods of Operations Research*, 60(1), 1–13.

Sun, X. (2020). *Facilitating Cooperative Truck Platooning for Energy Savings: Path Planning, Platoon Formation and Benefit Redistribution*.

Sutton, R. S., Barto, A. G., et al. (1998). *Introduction to reinforcement learning*, volume 135. MIT press Cambridge.

Sykora, H. T., Sadeghpour, M., Ge, J. I., Bachrathy, D., & Orosz, G. (2020). On the moment dynamics of stochastically delayed linear control systems. *International Journal of Robust and Nonlinear Control*, 30(18), 8074–8097.

Tang, L., Tan, Q., Shi, Y., Wang, C., & Chen, Q. (2018). Adaptive virtual resource allocation in 5g network slicing using constrained markov decision process. *IEEE Access*, 6, 61184–61195. https://doi.org/10.1109/ACCESS.2018.2876544

Tipireddy, R., Chatterjee, S., Paulson, P., Oster, M., & Halappanavar, M. (2017). Agent-centric approach for cybersecurity decision-support with partial observability. *2017 IEEE International Symposium on Technologies for Homeland Security (HST)*, 1–6. https://doi.org/10.1109/THS.2017.7943478

Treiber, M., Hennecke, A., & Helbing, D. (2000). Congested traffic states in empirical observations and microscopic simulations. *Physical review E*, 62(2), 1805.

Treiber, M. & Kesting, A. (2012). *Traffic Flow Dynamics: Data, Models and Simulation*. Springer Science & Business Media.

Trippel, T., Weisse, O., Xu, W., Honeyman, P., & Fu, K. (2017). Walnut: Waging doubt on the integrity of mems accelerometers with acoustic injection attacks. *Security and Privacy (EuroS&P), 2017 IEEE European Symposium on*, 3–18.

Uchiya, T., Nakamura, A., & Kudo, M. (2010). Algorithms for adversarial bandit problems with multiple plays. *Algorithmic Learning Theory*, 375–389.

USDOT (2016). *Connected vehicles and cyber security*. https://www.its.dot.gov/factsheets/pdf/cv_%20cybersecurity.pdf. Accessed February 16 2022

Van Arem, B., Van Driel, C. J., & Visser, R. (2006). The impact of cooperative adaptive cruise control on traffic-flow characteristics. *IEEE Transactions on intelligent transportation systems*, 7(4), 429–436.

van Wyk, F., Khojandi, A., & Masoud, N. (2019). A path towards understanding factors affecting crash severity in autonomous vehicles using current naturalistic driving data. *Proceedings of SAI Intelligent Systems Conference*, 106–120.

van Wyk, F., Khojandi, A., & Masoud, N. (2020). Optimal switching policy between driving entities in semi-autonomous vehicles. *Transportation Research Part C: Emerging Technologies*, 114, 517–531.

Van Wyk, F., Wang, Y., Khojandi, A., & Masoud, N. (2019). Real-time sensor anomaly detection and identification in automated vehicles. *IEEE Transactions on Intelligent Transportation Systems*, 21(3), 1264–1276.

Vengerov, D. (2007). A reinforcement learning approach to dynamic resource allocation. *Engineering Applications of Artificial Intelligence*, 20(3), 383–390. https://doi.org/https://doi.org/10.1016/j.engappai.2006.06.019

Vidal, R., Shakernia, O., Kim, H. J., Shim, D. H., & Sastry, S. (2002). Probabilistic pursuit-evasion games: theory, implementation, and experimental evaluation. *IEEE transactions on robotics and automation*, 18(5), 662–669.

Wan, E. (2006). Sigma-point filters: An overview with applications to integrated navigation and vision assisted control. *Nonlinear Statistical Signal Processing Workshop, 2006 IEEE*, 201–202.

Wang, P., Wu, X., & He, X. (2020a). Modeling and analyzing cyberattack effects on connected automated vehicular platoons. *Transportation research part C: emerging technologies*, 115, 102625.

Wang, Q. & Liu, M. (2015). Learning in hide-and-seek. *IEEE/ACM transactions on networking*, 24(2), 1279–1292.

Wang, Y. & Masoud, N. (2021). Adversarial online learning with variable plays in the pursuit-evasion game: Theoretical foundations and application in connected and automated vehicle cybersecurity. *IEEE Access*, 9, 142475–142488.

Wang, Y. & Masoud, N. (N.D.). Dynamic security resource allocation for connected and automated vehicles. *To be submitted.*

Wang, Y., Masoud, N., & Khojandi, A. (2020b). Anomaly detection in connected and automated vehicles using an augmented state formulation. *2020 Forum on Integrated and Sustainable Transportation Systems (FISTS)*, 156–161.

Wang, Y., Masoud, N., & Khojandi, A. (2020c). Real-time sensor anomaly detection and recovery in connected automated vehicle sensors. *IEEE transactions on intelligent transportation systems*, 22(3), 1411–1421.

Wang, Y., Zhang, R., Masoud, N., & Liu, H. X. (N.D.). Anomaly detection and string stability analysis in connected automated vehicular platoons. *Under first round of review*.

Watts, J., van Wyk, F., Rezaei, S., Wang, Y., Masoud, N., & Khojandi, A. (2022). A dynamic deep reinforcement learning-bayesian framework for anomaly detection. *IEEE Transactions on Intelligent Transportation Systems*.

Weimerskirch, A. & Gaynier, R. (2015). An overview of automotive cybersecurity: Challenges and solution approaches. *TrustED@ CCS*, 53.

Wünnenberg, J. & Frank, P. (1987). Sensor fault detection via robust observers. *System fault diagnostics, reliability and related knowledge-based approaches*, 147–160. Springer.

Xia, Y., Qin, T., Ma, W., Yu, N., & Liu, T.-Y. (2016). Budgeted multi-armed bandits with multiple plays. *IJCAI*, 2210–2216.

Xiao, Y., Codevilla, F., Gurram, A., Urfalioglu, O., & López, A. M. (2020). Multimodal end-to-end autonomous driving. *IEEE Transactions on Intelligent Transportation Systems*.

Xiao, Z., Song, W., & Chen, Q. (2013). Dynamic resource allocation using virtual machines for cloud computing environment. *IEEE Transactions on Parallel and Distributed Systems*, 24(6), 1107–1117. https://doi.org/10.1109/TPDS.2012.283

Yan, C., Xu, W., & Liu, J. (2016). Can you trust autonomous vehicles: Contactless attacks against sensors of self-driving vehicle. *DEF CON*, 24.

Yang, T. & Lv, C. (2021). A secure sensor fusion framework for connected and automated vehicles under sensor attacks. *IEEE Internet of Things Journal*.

Zhang, E., Masoud, N., Bandegi, M., & Malhan, R. K. (2022). Predicting risky driving in a connected vehicle environment. *IEEE Transactions on Intelligent Transportation Systems*.

Zhang, L. & Orosz, G. (2016). Motif-based design for connected vehicle systems in presence of heterogeneous connectivity structures and time delays. *IEEE Transactions on Intelligent Transportation Systems*, 17(6), 1638–1651.

Zhang, R., Liang, Y.-c., & Cui, S. (2010). Dynamic resource allocation in cognitive radio networks. *IEEE Signal Processing Magazine*, 27(3), 102–114. https://doi.org/10.1109/MSP.2010.936022