DEGREE PROJECT IN ELECTRICAL ENGINEERING,
SECOND CYCLE, 30 CREDITS
STOCKHOLM, SWEDEN 2021

VePMAD: A Vehicular Platoon Management Anomaly Detection System

A Case Study of Car-following Mode, Middle Join and Exit Maneuvers

WEAAM BAYAA

KTH ROYAL INSTITUTE OF TECHNOLOGY
SCHOOL OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Abstract

Vehicle communication using sensors and wireless channels plays an important role to allow exchanging information. Adding more components to allow exchanging more information with infrastructure enhanced the capabilities of vehicles and enabled the rise of Cooperative Intelligent Transport Systems (C-ITS). Leveraging such capabilities, more applications such as Cooperative Adaptive Cruise Control (CACC) and platooning were introduced. CACC is an enhancement of Adaptive Cruise Control (ACC). It enables longitudinal automated vehicle control and follows the Constant Time Gap (CTG) strategy where, distance between vehicles is proportional to the speed. Platooning is different in terms of addressing both longitudinal and lateral control. In addition, it adopts the Constant Distance Gap (CDG) control strategy, with separation between vehicles unchanged with speed. Platooning requires close coupling and accordingly achieves goals of increased lane throughput and reduced energy consumption. When a longitudinal controller only is used, platooning operates in car-following mode and no Platoon Management Protocol (PMP) is used. On the other hand, when both longitudinal and lateral controllers are used, platooning operates in maneuver mode and coordination between vehicles is needed to perform maneuvers. Exchanging information allows the platoon to make real time maneuvering decisions. However, all the aforementioned benefits of platooning cannot be achieved if the system is vulnerable to misbehavior (i.e., the platoon is behaving incorrectly). Most of work in the literature attributes this misbehavior to malicious actors where an attacker injects malicious messages. Standards made efforts to develop security services to authenticate and authorize the sender. However, authenticated users equipped with cryptographic primitives can mount attacks (i.e., falsification attacks) and accordingly they cannot be detected by standard services such as cryptographic signatures. Misbehavior can disturb platoon behavior or even cause collision. Many Misbehavior Detection Schemes (MDSs) are proposed in the literature in the context of Vehicular ad hoc network (VANET) and CACC. These MDSs apply algorithms or rules to detect sudden or gradual changes of kinematic information disseminated by other vehicles. Reusing these MDSs directly during maneuvers can lead to false positives when they treat changes in kinematic information during the maneuver as an attack. This thesis addresses this gap by designing a new modular framework that has the capability to discern maneuvering process from misbehavior by leveraging platoon behavior recognition, that is, the platoon mode of operation (e.g., car-following mode or maneuver mode). In addition, it has the ability to recognize the undergoing maneuver (e.g., middle join or exit). Based on the platoon behavior recognition module, the anomaly

detection module detects deviations from expected behavior. Unsupervised machine learning, notably Hidden Markov Model with Gaussian Mixture Model emission (GMMHMM), is used to learn the nominal behavior of the platoon during different modes and maneuvers. This is used later by the platoon behavior recognition and anomaly detection modules. GMMHMM is trained with nominal behavior of platoon using multivariate time series representing kinematic characteristics of the vehicles. Different models are used to detect attacks in different scenarios (e.g., different speeds). Two approaches for anomaly detection are investigated, Viterbi algorithm based anomaly detection and Forward algorithm based anomaly detection. The proposed framework managed to detect misbehavior whether the compromised vehicle is a platoon leader or follower. Empirical results show very high performance, with the platoon behavior recognition module reaching 100% in terms of accuracy. In addition, it can predict ongoing platoon behavior at early stages and accordingly, use the correct model representing the nominal behavior. Forward algorithm based anomaly detection, which rely on computing likelihood, showed better performance reaching 98% with slight variations in terms of accuracy, precision, recall and F1 score. Different platooning controllers can be resilient to some attacks and accordingly, the attack can result in slight deviation from nominal behavior. However, The anomaly detection module was able to detect this deviation.

Keywords: MDS, GMMHMM, PMP, Machine Learning, Bayesian information Criterion (BIC), Platoon Behavior Recognition

Sammanfattning

Kommunikation mellan fordon som använder sensorer och radiokommunikation spelar en viktig roll för att kunna möjliggöra informationsutbyte. Genom att lägga till fler komponenter för infrastrukturkommunikation förbättras fordonens generella kommunikationskapacitet och möjliggör C-ITS. Det möjliggör också för att introducera ytterligare applikationer, exempelvis CACC samt plutonering. CACC är en förbättring av ACC -konceptet. Denna teknik möjliggör longitudinell automatiserad fordonskontroll och följer en CTG -strategi där avståndet mellan fordon är proportionellt mot hastigheten. Plutonering är annorlunda med avseende på att hantera longitudinell och lateral kontroll. Dessutom antar den en kontrollstrategi för CDG där avståndet mellan fordon förblir oförändrat med hastighet. Plutonering kräver en nära koppling mellan fordon för att uppnå målet med ökad filgenomströmning och reducerad energikonsumtion. När enbart longitudinell kontroll är aktiverad, fungerar plutonering i bilföljande läge och funktionen PMP används inte. När både longitudinella och laterala kontroller används, arbetar plutonen istället i manöverläge och samordning mellan fordon behövs för att utföra olika manövrar. Informationsutbytet möjliggör att plutonen kan man manövrera i realtid. Alla ovan nämnda fördelar med plutonering kan emellertid inte uppnås om systemet är sårbart för felbeteende, det vill säga att plutonen beter sig fel. I litteraturen kopplas detta missförhållande till skadliga aktörer där en angripare injicerar skadliga meddelanden. I standardiseringsarbeten har man försökt utveckla säkerhetstjänster för att autentisera och auktorisera avsändaren. Trots detta kan autentiserade användare utrustade med kryptografiska primitiv upprätta förfalskningsattacker som inte detekteras av standardtjänster som kryptografiska signaturer. Felaktigt handhavande kan orsaka störningar i plutonens beteende eller till och med orsaka kollisioner och följaktligen påverka tillförlitligheten. Det finns manga MDSs beskrivna i litteraturen i relation till VANET och CACC. MDSs använder algoritmer eller regler för att detektera snabba eller långsamma förändringar kinematisk information som sprids av andra fordon. Direkt använding av MDSs under manövrar kan leda till falska positiva resultat eftersom de kommer att behandla förändringar i kinematisk information under manövern som en attack. Denna avhandling adresserar detta gap genom utformningen av ett modulärt ramverk som kan urskilja manöverprocessen från misskötsamhet genom att utnyttja plutonens beteendeigenkänningsmodul för att intelligent känna igen plutonläget (t.ex. bilföljande läge eller manöverläge). Ramverket har vidare egenskapen att känna igen pågående manövrar (frikoppling eller växelbyte) och avvikelser från förväntat beteende. Modulen använder en oövervakad maskininlärningssmodell, GMMHMM, för att lära en plutons normala beteende under olika lägen och manövrar som sedan används

för plutonbeteendeigenkänning och avvikelsedetektion. GMMHMM tränas på data från plutoneringens normalbeteende i form av multivariata tidsserier som representerar fordonets kinematiska karakteristik. Olika modeller används för att upptäcka attacker i olika scenarier (t.ex. olika hastigheter). Två tillvägagångssätt för avvikelsedetektion undersöks, Viterbi-algoritmen samt Forward-algoritmen. Det föreslagna systemet lyckas upptäcka det felaktiga beteendet oavsett om det komprometterade fordonet är en plutonledare eller följare. Empiriska resultat visar mycket hög prestanda för beteendeigenkänningsmodulen som når 100%. Dessutom kan den känna igen plutonens beteende i ett tidigt skede. Resultat med Forward- algoritmen för avvikelsedetektion visar på en prestanda på 98% med små variationer med avseende på måtten accuracy, precision, recall och F1-score. Avvikelsedetektionsmodulen kan även upptäcka små avvikelser i beteende.

Nyckelord: MDS, GMMHMM, PMP, Maskininlärnings, BIC, Plutonbeteendeigenkänning

Acknowledgments

I would like to thank all people who helped me during this period including my family, my colleagues and of course my supervisor, Mohammad Khodaei, for his supervision, patience, guidance and constant help and support during this work. I would like to thank Konstantinos Kalogiannis who provided continuous support and dataset used in this thesis and Prof. Panos Papadimitratos for fruitful discussions as well as his feedback on this thesis.

Contents

List of Figures

List of Tables

Acronym

3GPP 3rd Generation Partnership Project

ACC Adaptive Cruise Control

AUC Area Under the Curve

AUC-ROC Area Under the Curve and Receiver Operator Characteristic

BIC Bayesian information Criterion

BSM Basic Safety Messages

C-ITS Cooperative Intelligent Transport Systems

CA Certificate Authority

CACC Cooperative Adaptive Cruise Control

CAM Cooperative Awareness Messages

CAN Controller Area Network

CC Cruise Control

CDG Constant Distance Gap

CHMM Continuous Hidden Markov Model

CPS Cyber Physical Systems

CRL Certificate Revocation List

CTG Constant Time Gap

CVS Constant Vehicle Spacing

DENM Decentralized Environmental Notification Message

DHMM Discrete Hidden Markov Model

DoS Denial of Service

DSRC Dedicated Short Range Communication

ECDSA Elliptic Curve Digital Signature Algorithm

EKF Extended Kalman Filter

ETSI European Telecommunications Standards Institute

FN False Negative

FP False Positive

FPR False Positive Rate

GMM Gaussian Mixture Model

GMMHMM Hidden Markov Model with Gaussian Mixture Model emission

HMM Hidden Markov Model

HSM Hardware Security Module

IEEE Institute of Electrical and Electronics Engineers

ITS Intelligent Transport Systems

LiDAR Light Detection and Ranging

LIN Local Interconnect Network

LSTM Long Short-Term Memory

MAC Message Authentication Code

MANET Mobile ad hoc network

MDS Misbehavior Detection Scheme

MMSE Minimum Mean Squared Error

ML Machine Learning

MLP Multi-Layer Perceptron

OBU On Board Unit

PC Pseudonym Certificate

PCM PLATOON CONTROL MESSAGE

PKI Public Key Infrastructure

PMP Platoon Management Protocol

RSU Road Side Unit

SAE Society of Automotive Engineers

SCMS Security Credentials Management System

TN True Negative

TP True Positive

V2X Vehicle to Everything

V2V Vehicle to Vehicle

V2I Vehicle to Infrastructure

V2P Vehicle to Pedestrian

V2C Vehicle to Cloud

VANET Vehicular ad hoc network

VC Vehicular Communication

VCPS Vehicular Cyber Physical Systems

VLC Visible light communication

VPKI Vehicular Public Key Infrastructure

WAVE Wireless Access in Vehicular Environments

WSU Wireless Safety Unit

Chapter 1

Introduction

The aim of this chapter is to provide an overview of cooperative intelligent transport systems. This overview will be used as a foundation for the introduction of the research questions addressed in this thesis and problem statement.

1.1 Background

Technology development in areas such as wireless communication, computing systems and remote sensing enabled advancement of Intelligent Transport Systems (ITS). Appropriate Vehicular Communication (VC) architecture allowed exchanging information between vehicles and other components in the system such as Road Side Units (RSUs) [2]. Future automated vehicles will rely on Vehicle to Everything (V2X) communication to enhance road safety and efficiency. V2X communications include Vehicle to Vehicle (V2V), Vehicle to Infrastructure (V2I), Vehicle to Pedestrian (V2P) and Vehicle to Cloud (V2C). Applying Mobile ad hoc network (MANET) principles to vehicles resulted in the introduction of Vehicular ad hoc network (VANET). VANET became a key part of ITS framework to allow vehicles to communicate with each other using messages that could contain information about road conditions such as traffic congestion and accidents. These benefits can be achieved if these road related messages are genuine and reliable. Main characteristics of VANET are high node mobility and reliance on message contents, including position information. Recently, more vehicles will be equipped with GPS and WiFi devices which enable V2X communication and form a VANET. The goal of VANET architecture is to allow the communication among surrounding vehicles and between vehicles and infrastructure. Two message types used in European model: periodic based message called Cooperative Awareness Messages (CAM) [3] and event-driven one called Decentralized Environmental Notification Message (DENM) [4]. In US model, two-part message called Basic Safety Messages (BSM) [5] is used.

The term Cyber Physical Systemss (CPSs) has been used to describe life-critical systems that are characterized by large-scale and geographically dispersed deployment of networked devices equipped with sensors, actuators, control and networking components [6]. Accordingly, VANET system can be considered as CPS. Vehicular Cyber Physical Systems (VCPS) is an example of CPS in which vehicles communicate via vehicular networking. This system consists of other

components that vehicles interact with such as RSUs and back-end systems. One application of such system is platoon based VCPS in which vehicles are driven in a platoon-based pattern, with a closed feedback loop between the cyber plane and physical plane. The cyber plane describes the computing and communications aspects of the vehicle while physical plane describes vehicle's physical dynamics (e.g., mobility) considering traffic environment. Tight integration between the two planes is required to achieve the stability and efficiency of platoon-based VCPS.

In practice, a platoon is formed by grouping consecutive vehicles with close space and moving in the same direction. Platoon-based VCPS can guarantee improved road safety, while increasing the infrastructure usage and reducing fuel consumption. Reduction of distance headways between vehicles will contribute to increase road capacity. On the other hand, reduction of unnecessary velocity changes and aerodynamic drag on following vehicles will contribute to reduction of energy consumption and exhaust emissions. It is expected to have different types of vehicles in highways (e.g., Platoon-enabled vehicle and other vehicles which don't have the capability of platooning). Platoon-enabled vehicles have all required hardware, software and subscriptions in platooning service. Platoon-enabled vehicle can be either platooned or non platooned. A platooned vehicle is a member of platoon and can be either leader or follower. While the platoon drive on the highway, a maneuver can start at any point taking into consideration that only one maneuver can be conducted at a time to avoid complex coordination. Platoon Management Protocol (PMP) and its operations are discussed in [7] to allow platoon maneuvers. A basic functionality in platoon-based VCPS involves platoon formation, merging and splitting, etc., [8]. According to platoon pre-standardization [9], CAM can be used to indicate the presence of platoon-enabled vehicle. A new container called "PlatooningContainer" is added to CAM to carry information about vehicle and its ability of platooning. The standard can allow backward compatibility with legacy CAM which means that platoon-enabled vehicles can read this container while other vehicles can ignore it.

1.2 Motivation

Attacks in platoon-based VCPS can have huge impact as they can be tailored to cause real-world harm or loss of life. Accordingly, it is crucial to secure these systems and thus ensure safe operation of these systems. Authors in [10] illustrated vulnerabilities in a three layer framework (sensing, communication and control). Sensing includes all types of threats that will impact the functionality of vehicle sensors. Communication includes both intra-veihcle and inter-vehicle communications which can be vulnerable to attacks either from sensing layer targeting the internal vehicle network, such as Controller Area Network (CAN), Local Interconnect Network (LIN), etc., or external networks, i.e., V2X communication. Threats to both the sensing and communication layers can affect the control layer. Accordingly, security solutions must work together at all layers down to hardware to ensure secure operation of the system. Authors in [11] concluded that communication is more effective than distance sensors in terms of platoon safety. Moreover, information contained in event data such as drivers' braking events can be more effective for platoon management than some traditional information such as distance and vehicle speed. One of the security

challenges in platoon-based VCPS is to ensure the correctness of exchanged information in inter-vehicle communication.

Many mechanisms and approaches are proposed which can be categorized into proactive and reactive mechanisms based on their primary mode of operation [12]. Proactive mechanisms are aiming to apply preventive mechanisms to restrict access to platoon-based VCPS. These mechanisms are aiming to protect the system from external attackers by enforcing security policies such as integrity and authenticity checks through the use of a Vehicular Public Key Infrastructure (VPKI), e.g., [13], [14], [15], [16], [17], [18], [19]. If any vehicle disseminated messages with non-valid cryptographic signatures, it will be excluded from the system. However, if the attacker managed to obtain a valid signature, attacks will be successful. Therefore, reactive mechanisms are essential since they aim to detect and react to attacks which are not captured or prevented by proactive mechanisms (i.e., misbehavior). These mechanisms are aiming to detect internal attackers which are authenticated vehicles that can communicate with other vehicles [20], [21].

In [22], authors clarified that misbehavior includes both malicious participants and faulty nodes. Although both are introducing incorrect and inaccurate data, malicious nodes are doing this intentionally. Moreover, they derived three categories of Misbehavior Detection Schemes (MDSs) in terms of scope of detection: local, cooperative and global. Local MDS measures correctness of messages by checking internal consistency and optionally vehicles' sensors. Cooperative MDS relies on cooperation between vehicles and possibly RSUs to detect a misbehavior. Global MDS relies, at least partially, on back-end systems. In this thesis, only malicious nodes are considered as a source of introduction of incorrect data. Furthermore, local MDS as a reactive mechanism is used for detecting these malicious nodes. In existing work, [23], [24], [25] and [26] proposed MDSs assume that all vehicles in platoon are driving in a single straight lane and evaluate interactions between platoon members to ensure that all vehicles behave according to prescribed control laws and within expected performance bounds. In order to enable large deployment of platooning, it's essential that all possible attack scenarios (e.g., attacks during maneuver) are also considered.

1.3 Problem Statement

Inter-vehicle communications play an important role in platoon-based VCPS. Accordingly, interest in research increased to secure it against malicious messages that can threaten the safe operations of these systems [27]. Adopting existing solutions will not lead to the correct detection of misbehavior when PMP is considered. The main challenge lies in correctly identifying the misbehavior during platoon operations (e.g., car-following mode and maneuver mode) by equipping the MDS with capabilities to distinguish between normal operation of platoon and misbehavior. This thesis is concerned with the following high level research questions.

RQ1: By considering PMP that controls platooning operations and maneuvers, what aspects need to be considered while designing a local MDS deployed in each vehicle? This is the primary research question of this thesis, asking what extra design requirements beyond car-following mode that need to be considered

in order to avoid degrading the performance of local MDS which is not designed to consider platooning management operations.

RQ2: How to design an MDS that can take into consideration different maneuvers in PMP?

RQ3: Can MDS be reused for different platooning controllers or new MDS is needed for each controller?

RQ4: How to design the MDS in a way that enables it to recognize the undergoing maneuver and intelligently detect deviations from expected behavior?

1.4 Contributions

A state-of-the-art MDS that can take into consideration the dynamic environment in the decision making is introduced. Leveraging machine learning algorithms, the MDS has the ability to recognize the mode of the operation of the platoon (car-following or maneuver mode) at early stage and predict the misbehavior. Unlike existing MDSs, proposed framework has the ability to discern misbehavior from maneuver operation and accordingly, avoid treating a maneuver as a misbehavior. In addition, the framework is extensible in terms of maneuvers and controllers. In this thesis, only platooning controllers are considered (e.g., PATH and Flatbed) but the framework can be extended to include Cooperative Adaptive Cruise Control (CACC) controllers (e.g., Consensus and Ploeg). In addition, only middle join and exit maneuvers are considered but the framework can be extended to include other maneuvers (e.g., tail join).

1.5 Thesis Structure

The thesis is structured as follows, Chapter 2 reviews related work with explanation of technical background of platooning. In addition, survey of related standards are outlined. Chapter 3 is dedicated to describe the system model, threat model and attacks used for evaluating the proposed framework. Chapter 4 provides deeper explanation of the methodology and description of the implementation of the methodology. Results are presented and discussed in Chapter 5. The last chapter, Chapter 6, is dedicated to outline conclusions from obtained results and possible future work and development.

Chapter 2

Related Work

This chapter introduces the concept of platooning, highlighting the difference between platoon and CACC, followed by a detailed description of platooning system model. Assumptions regarding existing security architecture based on existing standards are highlighted. Existing platooning attacker models are reviewed alongside existing misbehavior detection mechanisms. In addition, evaluation of existing detectors limitations in literature are discussed.

2.1 Platooning System Model

In this section, an overview about platooning is introduced. In addition, platooning controllers as well as key enabling standards for Cooperative Intelligent Transport Systems (C-ITS) and platooning are outlined.

2.1.1 Platooning Overview

Cruise Control (CC) systems are available commercially in many vehicles to regulate vehicle speed. Recently, Adaptive Cruise Control (ACC) systems became commercially available in high-end vehicles to maintain a preset following gap to the preceding vehicle. ACC system uses radar or Light Detection and Ranging (LiDAR) sensors to sense the relative distance to preceding vehicles and use it to generate throttle or break command to maintain the gap [28]. As an evolution to ACC, Cooperative Adaptive Cruise Control (CACC) system incorporates exchanging information between vehicles using V2V communication taking advantage of the development in wireless communication technology such as Dedicated Short Range Communication (DSRC). In ACC and CACC, vehicles uses longitudinal control to control the throttle and break based on received information from other vehicles. However, lateral control which is the steering of the vehicle is the responsibility of the driver. The term CACC has been used loosely in literature and is often mistakenly assumed to be synonymous with platooning [29]. European Telecommunications Standards Institute (ETSI) platooning pre-standardization, (ETSI TR 103 298) [9], distinguished between the two terms in two ways. First, only vehicle speed control will be automated in CACC addressing longitudinal control while platooning is addressing both longitudinal and lateral control. In other words, CACC represents automation

level 1 of automation in Society of Automotive Engineers (SAE) while platooning represents at least automation level 2. Second difference is the reliance of platooning systems on Constant Distance Gap (CDG) control strategy where separation between vehicles remains unchanged with speed while CACC control strategy is based on a Constant Time Gap (CTG) where distance between vehicles is proportional to the speed [30]. An aspect that is common between platooning and CACC, both are enabled by leveraging V2X communication and accordingly if V2X communication is not present, platooning and CACC cannot work.

2.1.2 Controllers

Each vehicle contains two-layer controller. The upper level controller is responsible for maintaining stability during platooning through computing the desired acceleration. However, acceleration is not used as a control input and accordingly lower level controller is used to control the actuation to track the desired acceleration through the a throttle or brake actions. Upper level controller contains both ACC and CACC control functions. To enable platooning formation and maneuvers, lateral-longitudinal control systems are needed to execute desired trajectories. Combined lateral and longitudinal controller has different implementations and it is easily accessed by attackers through either sensors or Wireless Safety Unit (WSU). Four main cooperative algorithms for upper-level controllers are discussed in [25], namely, PATH, Consensus [31], Flatbed [32] and Ploeg [33]. PATH and Flatbed are using Constant Vehicle Spacing (CVS) policy which is equivalent to CDG strategy. Due time limitation, only CVS based controllers are considered in this thesis. These controllers are relying on both sensors and V2V for sharing information. For PATH, both speed and acceleration forgery can impact its operation and cause in most situations a crash. PATH depends on multiple variables and accordingly, it could be impacted by forged beacons. For speed falsification, PATH showed that it will crash regardless the speed or position of attacker. On the other hand, Flatbed showed low sensitivity to the speed falsification. It requires all platoon members to share a common speed value which shared by a leader. Accordingly, it is expected that attacks from compromised leader will have more severe impact [1].

2.1.3 C-ITS Key Enabling Standards

This section briefly outlines the most relevant standardization activities. C-ITS relies on both V2V and V2I communication to exchange information. Institute of Electrical and Electronics Engineers (IEEE) 802.11p is proposed and provided amendments and enhancements to existing IEEE 802.11 standard which implies small modifications to achieve a robust connection and a fast setup for moving vehicles. IEEE 802.11p standard allows the use of 5.9 GHz licensed band to enable both V2V and V2I communication with a communication range that is typically between 100 and 500 meters in ad-hoc basis, i.e., a direct communication between nodes without intermediate base station. It defines only the specifications for the basic physical and medium access control layers. IEEE 1609 is defined for higher layers (above the IEEE 802.11p physical and medium access control layers). The combination of the IEEE 802.11p and IEEE 1609 standards is generally known as Wireless Access in Vehicular

Environments (WAVE) [34]. Similar standardization activities are driven by the ETSI where it defines a reference architecture for cooperative vehicular communications including six main layers: application layer, facilities layer, networking and transport layer including GeoNetworking, medium access layer, management entity and security entity. Two types of messages are communicated between vehicles in European model, namely, cooperative awareness message CAM [3] and event-driven decentralized environmental notification message DENM [4]. A corresponding message to these messages in US is BSM [35] which consists of two parts. First part of BSM is sent periodically similar to CAM while second part is similar to DENM which is only transmitted when a specific event occurs.

2.1.4 Platooning Pre-standardization

ETSI initiated an attempt to define V2X protocol for realizing platooning in ETSI TR 103 298 [9]. It proposes the use of CAM standard to include *PatooningContainer* in *CamParameters* to carry information about vehicles and their capability of platooning. CAMs with *PatooningContainer* is prerequisite for Join or merge maneuver to happen. This doesn't violate backward compatibility to legacy CAM since only vehicles interested in reading *PatooningContainer* will read it while other vehicles will just ignore it. Join procedure is initiated by sending JOIN REQUEST to join the vehicle/platoon in front of it. Once a JOIN RESPONSE is received, the vehicle will transmit PLATOON CONTROL MESSAGE (PCM) and moves to PLATOON mode. JOIN RESPONSE contains an encryption key for encryption of PCMs. Joining procedure can take 30-50 ms and up to 1 second. During PLATOON mode, PCMs are sent every 50 ms and absence of them for a certain period of time means that platoon need to be dissolved and re-initiated again. Then the vehicle intends to leave the platoon will indicate in PCMs that it has the intention to leave.

2.2 C-ITS Security Architecture

In this section, the baseline of standardized security measures and a brief overview of security credential management and privacy considerations are outlined. Two main standards are proposed to specify how certificates are used to secure the authenticity and integrity of the data. These standards are ETSI TS 103 097 standard [9] in Europe and the IEEE 1609.2 standard [36] in the US. Each exchanged message is signed with sender's secret key. The secret key and certificate will be attached to the message to enable the receiver to check the authenticity of the message. Both standards suggest the use of Elliptic Curve Digital Signature Algorithm (ECDSA) cryptographic algorithm. Signature and certificate will ensure sender authenticity and message integrity. In addition, they will offer protection against attackers that transmit messages without key and certificate. However, this process will not ensure message correctness. If attacker managed to get access to key and certificate (e.g., extracting them from flash or hard disk of old communication unit), he can transfer it to other device and use to exchange messages with correct signature. In order to solve this problem, SAVECOM project [37] outlined basic principles and architecture for VC systems. Authors in [38] and [39] proposed an implementation of SAVECOM

including the use of trusted hardware, named, Hardware Security Module (HSM) to protect key material against outside access. HSM protects key material and offer cryptographic operations to applications. In order to sign a message, it needs to be sent to HSM and signature is provided in return and accordingly key material never leave HSM. This prevents the attacker from acquiring key material and use it to sign fabricated messages. However, attacker can send fabricated messages to HSM to be signed since HSM cannot distinguish if the provided message is genuine or manipulated [40]. In addition, attacker can either manipulate sensor readings, modify sensors hardware or even inject false readings in CAN bus. This put requirements to protect the system from correctly signed messages with invalid content [20], [21].

Another aspect that need to be considered is related to certificate management which require infrastructure to manage certificates in C-ITS. These certificates shall be provided by trusted third party mainly Enrollment Authority and Authorization Authority [41]. Enrollment is the main access control to C-ITS which authenticates vehicles and grant them access to C-ITS. Authorization Authority provides vehicles valid certificates to gain authorization to use other services in C-ITS. A standard VPKI [18], [19] has the responsibility to manage the creation, distribution, revocation and administration of certificates. A problem can raise if these certificates have long validity date since attackers can track individual vehicles by tracing received messages with attached location information. This results in raising privacy problem. Accordingly, long term certificates are replaced with short term identifiers called pseudonyms. Having multiple short lived pseudonyms opens other type of attacks called Sybil attacks where the attacker can possess multiple pseudonyms and use them to conduct different types of attacks. Different schemes proposed to tackle this issue such as VPKI in [18], [42] and [43] where on demand approach is used to issue unlinkable pseudonyms and offer Sybil free environment as claimed by authors. In addition, pre-standardization work in ETSI started (TR 103415) [44] for pseudonym change management. As a part of security management, misbehaving vehicles shall be evicted from the system and their certificates will be included in Certificate Revocation List (CRL) and distributed from root Certificate Authority (CA) to subordinate CAs [45], [46].

2.3 Misbehavior in Platooning

2.3.1 Studied Attacker Models in Literature

In [47], a Security Credentials Management System (SCMS) is assumed to be used. Certificate-Based Authentication system that uses a Public Key Infrastructure for certificate management is used. Pseudonym Certificates (PCs) are used to protect the privacy of vehicles. Vehicles can obtain PCs for a short period of time and is used for BSM authentication. SCMS signs each message sent by vehicles with a certificate and accordingly prevents attacker from falsifying messages from other vehicles but it cannot prevent malicious actor from obtaining a certificate and misbehave (e.g., through injecting fabricated data). In other words, attacker can authenticate himself to SCMS as a regular vehicle and then apply attacks on application level. Only application level exploitation is considered in the threat model and accordingly, attacks such as jamming, physical

attacks on sensors or controllers, Denial of Service (DoS) attacks and software bugs whether in the infrastructure or vehicles are not considered. It is assumed that the attacker is aware of the application logic and how to craft actions to manipulate that logic. In some attacks, it is assumed that the attacker is able to use radios that allow him to reach vehicles beyond typical vehicular radios. In [48], security infrastructure and security standard protocol are assumed to be in place. In addition, security solutions such as certificates and cryptographic keys are use to provide privacy and confidentiality, respectively. Both are assumed to be stored in HSM to protect them from tampering. However, an attacker can hack HSM either by physical manipulation of the vehicle during manufacturing or maintenance or by doing reverse engineering of old HSM. Attacks such as jamming, packet injection and channel overhearing are considered. Attackers are assumed to be either road side attackers or part of VANET but not part of the platoon itself. The attacker aims to destabilize the platoon without being affected. Attacker is assumed to be insider attackers and a member of the platoon.

2.4 Existing Solutions in Literature

Misbehavior detection has been studied in the context of VANET and C-ITS. In VANET context, MDS classification is provided in [49] based on the used approach for detection. Two dimensions are used to distinguish MDS approaches which resulted in four classes. First dimension is whether the focus on the data contained in the messages sent by vehicles or on the nodes sending these messages. The first is called data-centric MDSs while the latter is called node-centric MDSs. The second dimension is based on the analysis of messages whether from a single vehicle (autonomous) or multiple vehicles (collaborative). These two dimensions resulted into four classes of MDSs: behavioral, trust-based, consistency and plausibility.

In the context of securing platoon during maneuvers, less work has been done in this area. In [48], a new hybrid security protocol is proposed, namely SP-VLC, aiming to secure platoon maneuvers under different attacks including platoon maneuver attacks. The protocol combines both IEEE 802.11p and Visible light communication (VLC). IEEE 802.11p is used to ensure sufficient transmission coverage in case of unavailability of VLC while VLC is used to ensure successful data transmission during jamming attacks. In this work, road side attacker is assumed to transmit either fake maneuver request packet or a fake maneuver response packet. In SP-VLC hybrid protocol, VLC is used for secret key establishment to construct the initial secret key securely between each pair of consecutive platoon members. To ensure that the messages is sent by platoon member, authentication using Message Authentication Code (MAC) to encrypt the unique identifiers of vehicle and platoon, and packet sequence number with the secret key. Finally, a mechanism to switch to either transmission over both IEEE 802.11p and VLC or VLC only according to conducted attack. In [47], authors presented attacks that attempt to exploit the functionality of the PMP implementation. The attacker is participating in the protocol in a way that will allow passing certificate based authentication and the application logic. The defense mechanism relies on sending maneuvers requests to RSUs which in turn will verify it with relevant information. If all checks are verified, maneuver

approval will be sent to requestor. During validation process, platoon will be switched to safe mode where it moves within the safety speed limit, i.e., 20 mph. If rejection is received from a RSU, the maneuver will be aborted.

2.4.1 Evaluating Existing Detectors Limitations

Machine Learning (ML) models have been used in VC systems to detect misbehaving vehicles. Techniques relying on supervised learning [50], semi-supervised learning and unsupervised learning [51] have been adopted. Any linear ML model that relies on linear function for its prediction function cannot be applied if system dynamics and observation models are non-linear. Linear ML models are simple while non-linear ML models imply more computational complexity. An example is the application of non-linear activation functions such as sigmoid or ReLU functions in artificial neurons in neural networks. Authors in [52] adopted both techniques based on neural networks such as Multi-Layer Perceptron (MLP) and Long Short-Term Memory (LSTM) classifiers but these techniques have not been tested in platooning context. Kalman Filter [53] is another solution which is used widely as a state estimator from noisy sensors information. When it is employed, fusion of data from different sensors are used to produce an accurate estimation of the system state. Similar to ML models, Kalman Filter can be used when system dynamics and observation models are linear since it has the ability to compute Minimum Mean Squared Error (MMSE) estimate. Kalman Filter has been used in the context of CACC to detect injection attack by detecting increase of deviation of received measurements through V2V from Kalman Filter estimations [25]. Furthermore, [1] attempted to use Kalman Filter during middle join and exit maneuvers. Results have shown that Kalman Filter detected two vehicles as malicious during the normal operation of middle join maneuver. Similar behavior has been indicated for exit maneuver where benign vehicles have been detected as malicious by Kalman Filter. Application of Kalman Filter to non-linear systems is difficult and accordingly, certain extensions need to be done. Extended Kalman Filter (EKF) was proposed to solve the problem of estimating non linear systems by linearizing non linear models and accordingly, traditional Kalman filter equations can be used. In practice, this approach has two main drawbacks. First, if assumptions of local linearity is violated, unstable filters can be produced due to linearization. Second, in many applications it is difficult to derive Jacobian metrics and accordingly, it can lead to significant implementation difficulties [54]. In this thesis, Gaussian Mixture Model (GMM) is used to model the probability density of non linearity of vehicles' and platooning dynamics. Furthermore, probability calculated by GMM is used as observation probability in Hidden Markov Model (HMM). Then, Viterbi algorithm is used for decision according to observation probabilities calculated by GMM. Forward algorithm is also used as an alternative to Viterbi algorithm. More details are provided in Chapter 4.

Chapter 3

Attacker Model and System Model

3.1 Attacker Model

In this thesis, internal attacker is assumed which implies that the attacker is an authenticated member of the VC network that can communicate with platoon members. The attacker has malicious intentions and aims to either cause a crash within the platoon as a primary goal or destabilize the platoon as a secondary goal. Attacker is active and generate signals or inject packets to perform the attack. Local attacks are considered which means that attacker can conduct an attack in small geographical area such as a highway. Attacker can be a sole attacker or conduct the attack in collaborated fashion. Attacker can compromise either platoon leader or platoon member. However, Sybil attacks where attacker can obtain multiple pseudonyms (certified key pairs) are assumed to be handled by standardized methods which discussed in section 2.2. Standardized security mechanisms are assumed to be in place such as pseudonymous certificates to protect driver's privacy and storage of corresponding private keys in a HSM to protect it against tampering. Countermeasure on certificate level for Sybil attack such as VPKI based model [18] where it can limit the validity of pseudonyms is assumed. By compromising the vehicle, it is assumed that the attacker has full control of inputs of vehicles and has the ability to inject falsified beacons such as manipulating speed, acceleration and position of compromised vehicle. RSUs which can monitor and transmit vehicles' speed and position to platoon members are not assumed to be available or trusted since their physical location on side roads make them vulnerable for tampering.

3.2 System Model

Platoon-based VCPS consists of platoon leader which is the front vehicle of the platoon and followers that follow the leader. Platoon is assumed to drive only in highways. Impact on the platoon due to traversing steep uphill or downhill is not considered. System model is build under the assumption that platoon speed (e.g., leader speed) is constant and will not change during

the maneuver. Communication exchange in platooning is enabled by equipping each vehicle with On Board Unit (OBU). Vehicles within the platoon consume data received through IEEE 802.11p and sensors. Sensors transmission range is limited to two consecutive vehicles while vehicles communicate wirelessly using IEEE 802.11p to allow transmission over bigger range and accordingly, enabling scenarios such as transmitting data from platoon leader to other vehicles. To enable stable platooning operation and to support efficient maneuver operations of platooning, performance requirements should be met. In 3rd Generation Partnership Project (3GPP) TS 22.186 [55], such performance requirements for platooning are specified in terms of end to end latency, reliability and communication range. Both car-following and maneuvers modes are supported. Only two maneuver operations are supported, namely, middle join and exit. Platoon followers must inform platoon leader regarding their intent to perform a maneuver. Platoon leader coordinates all maneuvers of platooning. Only one maneuver is allowed at a certain time. Vehicles are equipped with other supporting systems that can help during maneuver operations. Radar is an example of such systems where wavelengths are used to perceive objects and movements. Platooning formation is based on common criteria (e.g., final destination or route) in order to reduce number of maneuvers. It is assumed that RSUs broadcast list of possible target platoons that vehicles can join. Detailed description of supported platooning maneuver operations are provided in following section.

3.2.1 Supported Platooning Maneuver Operations

Basic car-following mode relies on the longitudinal controller only. However, in order to perform maneuvers such as middle join and exit maneuvers, coordination between longitudinal and lateral controllers is required. Description of these maneuvers is provided according to [56], [9] and [1]

Middle Join Maneuver

When a vehicle intend to join a platoon, it sends a join request to platoon leader. Platoon leader grant permission to joiner if a set of requirements are met such as maximum number of vehicles in the platoon is not violated, joiner is in the communication range of the platoon and both joiner and platoon are moving in the same direction. Platoon leader informs the vehicle behind joining position to do a split and create a space. Split operation will let vehicle behind joining position suspend the use of radar information while the joiner do a lane change. Split operation is done by longitudinal controller. Once the space is created, vehicle behind joining position informs platoon leader. Then, platoon leader instructs the joiner to do a lane change. Joiner perform lane change using lateral controller, set correct ID in the platoon and and change its operation mode. In platooning pre-standardization, this action corresponds to setting the flag *isJoinable* to False. When the join maneuver is completed successfully, platoon leader informs other vehicles in platoon and vehicles behind joiner to increment their IDs and ensure that spacing imposed by controller is fulfilled.

Exit Maneuver

When a follower vehicle decides to exit the platoon, it will send a leave request to platoon leader. Platoon leader grant permission to that vehicle to leave the platoon if no other vehicle sent any other maneuver request. Platoon leader will instruct the follower vehicle of leaver to split. This is an important step because the vehicle behind the leaver must suspend the use of radar information while the leaver do a lane change. When leaver and the vehicle behind it finish splitting operation, the platoon leader grant permission to leaver to do lane change and change its operation mode. This action corresponds to setting the flag *isJoinable* to True. When the leaver perform the exit maneuver successfully, platoon leader informs other vehicles in platoon and vehicles behind leaver to decrement their IDs and accelerate to fulfill spacing imposed by controller.

Chapter 4

Misbehavior Detection for Platooning

This chapter provides an overview of VePMAD, the proposed framework which its functionality is centered around research questions outlined in Chapter 1.3. First, fundamental concepts of HMM and Hidden Markov Model with Gaussian Mixture Model emission (GMMHMM), modelling approach on which VePMAD's architecture is built, are introduced. Then, architecture components are introduced and discussed in details.

4.1 Fundamental Concepts of HMM and GMM-HMM

HMM has been widely used to model driving behavior due its powerful ability to describe dynamic processes. It is based on two stochastic processes: an observable process which represents the sequence of observations of the system and hidden process which can be indirectly inferred by analyzing the the sequence of observations. The latter process can be either modeled as a Discrete Hidden Markov Model (DHMM) to associate a continuous feature vector to a discrete random state or a Continuous Hidden Markov Model (CHMM) to model the observation distributions for the feature vectors by either a single Gaussian or mixture of Gaussians. In both cases, observations depends on hidden states. In driver intention recognition context, discrete recognizer is used to model each maneuver as a Markov model trained by a set of samples of a complete maneuver. Such recognizer can be used as a classifier of maneuver after it has been completed (i.e. to classify the current driver as sporty or defensive). On the other hand, continuous recognizer is used in more advanced driver assistant systems to identify the maneuver at early stage.

HMM is determined by number of N possible states S = 1, 2, . . . ,N and can be written in a compact form as:

$$\lambda = (\pi, A, B) \tag{4.1}$$

The initial transition vector, π, is a vector containing the probability for the state of being the first state of the sequence. The state transition probabilities,

A,is a matrix consisting of the probabilities of transitioning from state Si to state Sj. Observation probability distribution, B, represent the probability of observation being generated from the state Si. HMM provide algorithms to solve three types of problems [57].

- Problem 1: given a sequence of observations O and HMM model (λ), compute the probability that the observed sequence is represented by HMM model (λ). It is an evaluation problem where scoring is used to determine how well a model is matching the observation sequence. It is useful in scenarios where a model need to be chosen among several competing models to best match observations. In the context of this thesis, it is used in platoon behavior recognition. Maximum likelihood method is used to solve this problem.

- Problem 2: given the observation sequence O and model (λ), compute the optimal state sequence that best explains a given observed sequence. Here, the attempt is to uncover the optimal hidden states sequence that best models the observations. This problem is solved using Viterbi algorithm [58].

- Problem 3: find parameters of HMM model (λ) = (A, B, π) to maximize the fit to an observed sequence (i.e., maximize $P(O \mid \lambda)$. It is a training problem where observations are used to adjust model parameters and adapt them to these observations which called training data to create a model for observed behavior. This problem is solved using Baum-Welch algorithm [59]. It is an iterative algorithm for non-convex problems that aims for finding local maximum depending on initial parameters which need to be chosen appropriately.

In this thesis, CHMM is used by observing continuous signals such as position, velocity, and acceleration to be able to recognize the platoon behavior at early stage and accordingly detect misbehavior as early as possible. Given time series $O = \{o_1, o_2, ...o_T\}$ and hidden sequence $\{z = z_1, z_2, ..., z_T\}$, initial probabilities π_i are expressed as:

$$\pi_i = P(z_1 = s_i), i = 1, ..., N \tag{4.2}$$

and $a_{i,j}$, the state transition probability distribution is expressed as:

$$a_{i,j} = P(z_{t+1} = s_i | z_t = s_j) \tag{4.3}$$

where $s_i, s_j \in$ S, and i, j $\in \{1, ..., N\}$. The probability of the observations can be discrete or continuous. In discrete distributions, observations can belong to codebook V = {v1, ..., vk} and the probability is defined as:

$$b_i(o_t) = P(o_t = v_k | z_t = s_i) \tag{4.4}$$

where $1 \leq$ i $\leq$ N, t = 1, ..., T. In continuous distribution, observations follow a specific distribution, e.g., a Gaussian distribution or a mixture of multiple Gaussians. GMMHMM is HMM with GMM emission probability distribution B = $\{\text{bi}(\cdot)\}$ which means that observation probabilities of HMM states are modelled with GMMs (e.g., represent the emission distribution as a mixture of

multiple multivariate Gaussian densities). The emission probabilities are defined as:

$$b_i(o_t) = \sum_{m=1}^{M} C_{i,m}\mathcal{S}(o_t)|\mu_{i,m}, \Sigma_{i,m} \tag{4.5}$$

where $1 \leq \mathrm{i} \leq \mathrm{N}$, M is the number of Gaussians, μ is the mean, Σ is the covariance and $C_{i,m}$ is the mixture coefficient where $\sum_{m=1}^{M} C_{i,m} = 1$. If M is large enough, a mixture of M Gaussian densities can effectively model any probability density function. Large enough M can result in restricting the covariance metrices and get a good approximation of any probability density function and reduce number of parameters that need to be updated during Baum-Welch. The compact form in equation 4.1 can be written as:

$$\lambda = (\pi, A, C, \mu, \Sigma) \tag{4.6}$$

One problem during training of HMM model is the assumption of knowing HMM topology beforehand which is not a realistic assumption in real-world applications. For instant, in weather prediction application, three hidden states can be assumed (i.e., hot, mild or cold). However, in more complex applications such as platooning it is more difficult to determine number of hidden states for HMM or number of components in GMMHMM. Accordingly, a criterion is needed to choose a configuration among many possible mixture of configurations. Bayesian information Criterion (BIC) [60] is a commonly used criterion to balance between likelihood of data and number of free parameters.

GMMHMM is used in platoon behavior modelling by defining the appropriate hidden states for HMM. In the latent space, sequence of observations that represent each platoon behavior will be modelled and accordingly, each unique sequence of hidden states should map to a sequence of a specific platoon behavior (e.g., maneuver). In the context of modelling platoon behavior, equation 4.5 is used. The aim is to understand the difference between different platoon behavior (e.g., car-following mode and different maneuvers). Each platoon behavior will have a GMMHMM model (λ) trained separately based on observations of each behavior. During training process, model parameters will be optimized to represent the platoon behavior accurately. Let $\lambda = f(\pi, A, C, \mu, \Sigma)$ represent GMMHMM model with its complete parameters set, the likelihood of an observed sequence O can be computed as:

$$P(O|\lambda) = \sum_{allz} P(O|z, \lambda)P(z|\lambda) \tag{4.7}$$

For platoon behavior recognition, at each instant t, the likelihood estimation of observations is calculated for each GMMHMM model (λ) using the forward algorithm. Then, the most likely GMMHMM model (λ) is selected as the right label ($\hat{\lambda}$) as expressed in 4.8:

$$\hat{\lambda}(t) = \underset{\lambda}{\operatorname{argmax}}\, P(O \mid \lambda_i) \tag{4.8}$$

4.2 VePMAD: Vehicular Platoon Management Anomaly Detection Framework

High level description of main principles of the proposed framework is shown

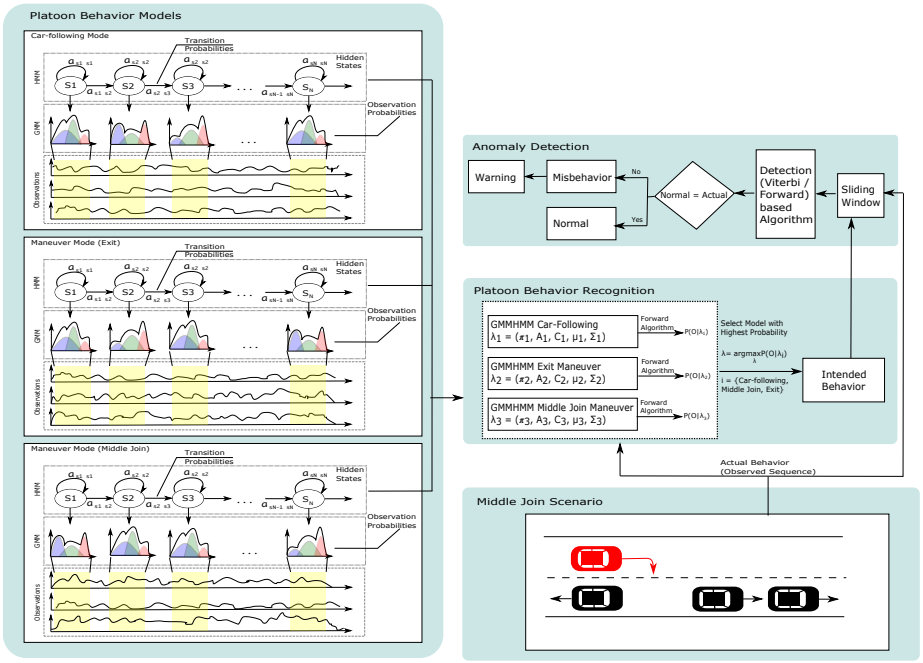


Figure 4.1: VePMAD Framework

in Figure 4.1. GMMHMMs are used to model the platoon behavior during car-following mode, middle join maneuver and exit maneuver. GMM is used to model the dependent relationship between kinematic information of vehicle (e.g., speed, acceleration and position) to describe the vehicle's behavior. Then, HMM is used to estimate the intended behavior of each vehicle in different modes (i.e., car-following or maneuver) based on the trained GMM. An example of middle join maneuver is shown in the figure where platoon behavior recognition module is employed to estimate the behavior in online manner. The observed sequence is evaluated by GMMHMM models. Forward algorithm is used to calculate the conditional probabilities. The intention corresponding to the largest value will be considered as platoon's intended behavior (i.e., middle join maneuver). The second underlying principle is that the anomaly detection module will use the model resulted from platoon behavior recognition module to estimate the normal behavior and compare it with actual behavior. Two approaches are used considering either comparing the hidden states named Viterbi-based, or comparing likelihood named forward-based. Any deviation from expected normal behavior will be reported as a misbehavior.

Details of creating GMMHMM models and using them in detecting misbehavior are shown in Figure 4.2. Two phases are indicated: training phase and inference phase. In training phase, GMMHMM models are trained offline in two steps. During first step, Baum-Welch algorithm is used to update the parameters by estimating GMM distribution parameters. One problem during training of HMM model is the assumption of knowing HMM topology beforehand which is not a realistic assumption in real-world applications. BIC is a criterion to select models among a finite set of models. It is based on likelihood function. BIC is a

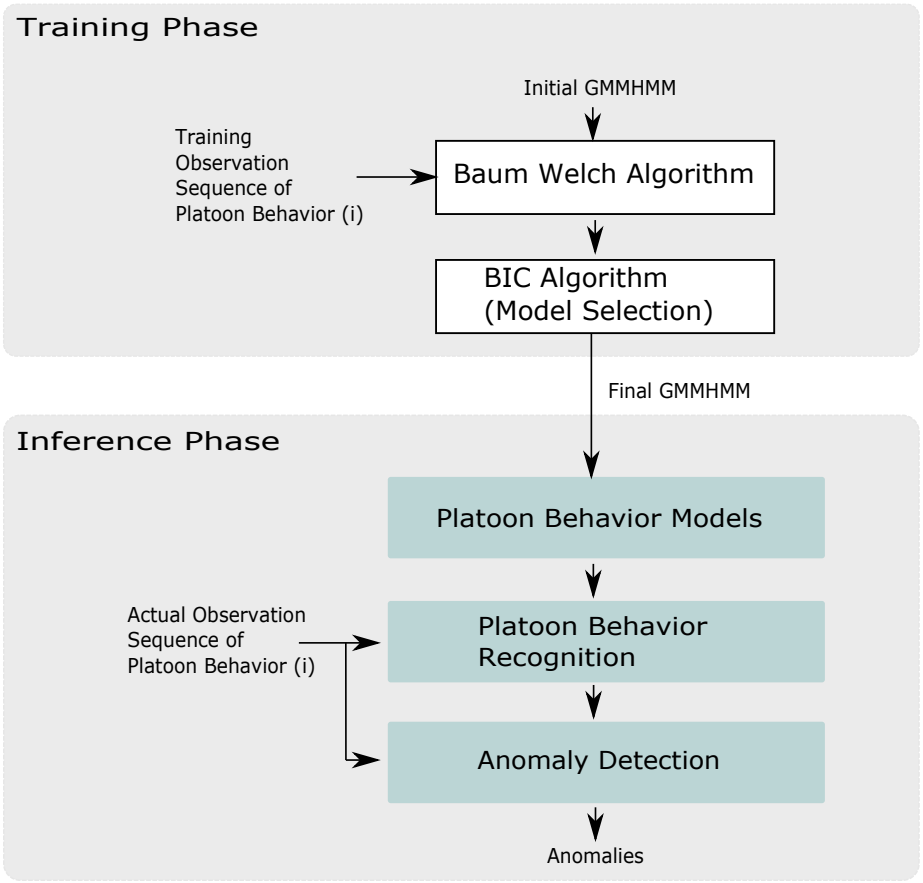


Figure 4.2: Training and Inference Phases

commonly used criterion to balance between likelihood of data and number of free parameters. When fitting models, it is possible to increase the likelihood by adding parameters but that would result in overfitting. BIC resolves this problem by introducing a penalty for the number of parameters in the model. In the second step BIC is repeated many times until the model parameters converge.

4.2.1 HMM and GMMHMM for Platoon Behavior Modelling

The goal platoon behavior modelling is to find the probability of sequence of observations in terms of acceleration, speed and position. During training, different data sets have been used representing each platoon behavior. Furthermore, it has been realized that controller behavior differs for the same maneuver and accordingly, a model per maneuver and per controller need to be trained. HMM need to be trained with labeled samples to determine model parameters (i.e., model grammar) to be used for two purposes, intention recognition and anomaly detection. The separate training process allows choosing number of states or components for each of the maneuver. Accordingly, observed features and their

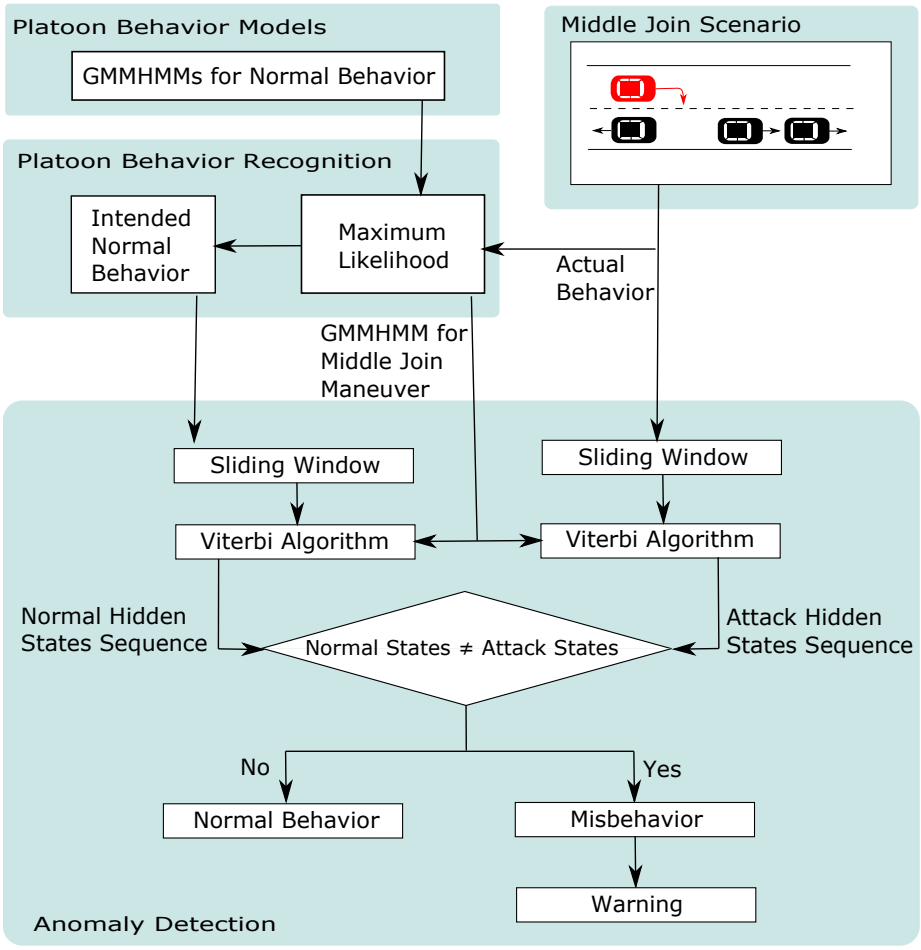


Figure 4.3: Viterbi-based Anomaly Detection

temporal dependencies will be used for fine-grained modelling which enhances the accuracy of mapping between the model's emissions and the observed features, especially when using Gaussian mixture distributions.

4.2.2 Anomaly Detection Approaches

Two approaches has been considered for anomaly detection. The first approach shown in Figure 4.3 is based on the interpretation of HMM as a method to find underlying sequence in latent space called hidden states which thought of as cause of the observations. Hence, observation sequence is replaced with a sequence of hidden states which makes it less complex and easier to compute with. After recognizing the undergoing maneuver, a sliding window is applied on both intended normal behavior and actual behavior. Then, Viterbi algorithm is applied on each sliding window and corresponding hidden states are determined. By comparing the hidden states sequence of normal behavior and hidden states sequence of attack, anomaly is computed.

The second approach shown in Figure 4.4 relies on adopting the same approach

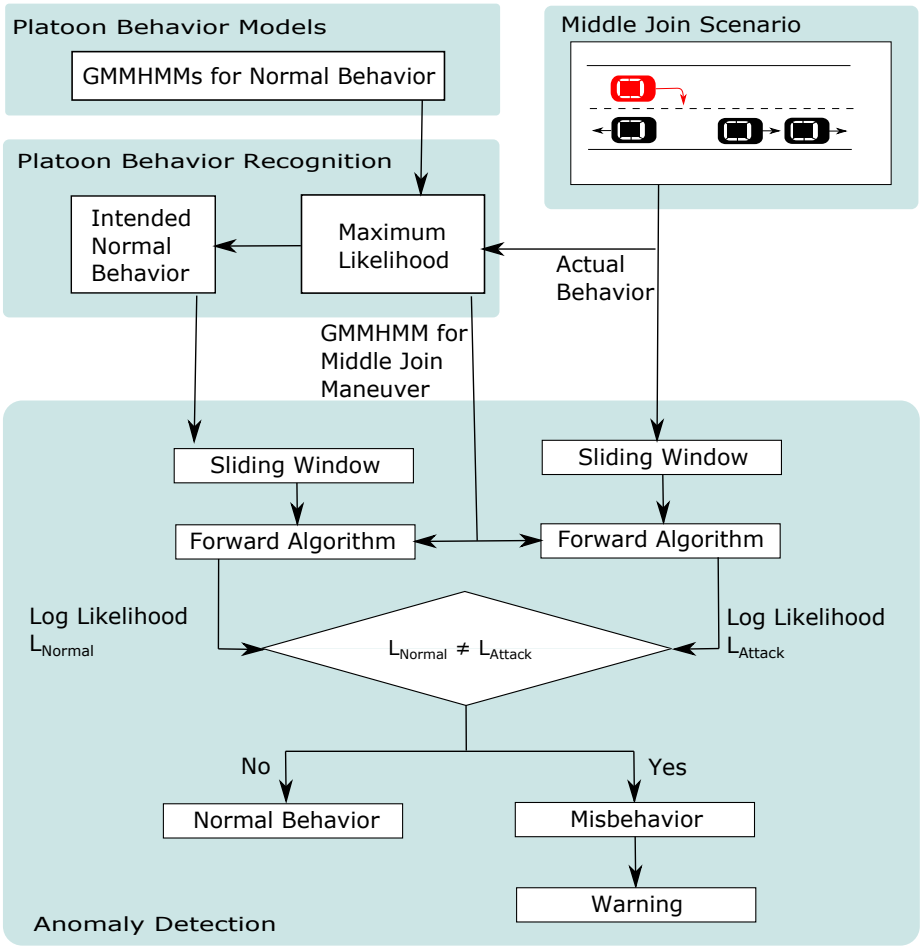


Figure 4.4: Forward-based Anomaly Detection

of applying same sliding window for both intended normal behavior and actual observations. Then, instead of determining the hidden states corresponding to each observations sequence, the likelihood of observation sequence of the attack is compared with likelihood of observation sequence of normal behavior. If any deviation is detected it will be counted as anomaly. Thus, probability distributions of two sequences representing normal behavior and attack are computed and two vectors of probability distributions are created for each GMMHMM. If these vectors are different, it is likely that their original event sequences are different indicating a misbehavior.

Chapter 5

Evaluation

This chapter describes the applied evaluation methodology. Both evaluation strategy and evaluation metrics are described in details. The performance of platoon behavior recognition and misbehavior detection components described in Chapter 4.2.1 and Chapter 4.2.2 are evaluated and discussed.

5.1 Metrics of Evaluation

In this section, evaluation metrics of proposed MDS are illustrated. Existing metrics in literature are reused such as accuracy and confusion matrix using False Positive (FP), False Negative (FN), True Positive (TP) and True Negative (TN). Detection accuracy is defined as the number of correct classifications (TP + TN) over all classifications (TP + FP + TN + FN). Other metrics such as precision which quantifies the relevance of detection events (TP/(TP + FP)) and recall which quantifies what rate of positives is actually detected (TP/(TP + FN)). Precision metric determines how precise and accurate the model is. In other words, it evaluates how many of these predicted positives are actually positive. It is a good metric to use if cost of false positives is high (e.g., non anomalous point is detected as anomaly). On the other hand, recall determines how many of actual positives the model managed to label them as true positives. Recall is a good metric to use if the cost of false negative is high (e.g., anomaly point is not detected). An optimal detector thus has a precision and a recall of 1. F1 score is also used and it represents the weighted average of precision and recall which means that it takes both false positives and false negatives into account. A good F1 score indicates that both false positives and false negatives are low and accordingly, real threats are correctly identified and MDS is not disturbed by false alarms. Area Under the Curve and Receiver Operator Characteristic (AUC-ROC) is used to measure the ability of classifier to distinguish between classes. The higher the Area Under the Curve (AUC), the better the performance of model at distinguishing positive and negative classes. Delay of detection is defined as the difference between detection time and time of attack.

5.2 Simulation Setup

Attacks and maneuvers implemented in [1] are leveraged to design and test the proposed framework. Plexe simulator (version 2.1) [61] is used which is based on VEINS [62] and extends SUMO [63]. Maneuvers included in Plexe have been extended to include middle join and exit maneuver for four controllers (Path, Consensus, Ploeg and Flatbed). In addition, Plexe is extended to include attacks. In this thesis, only CVS controllers are considered, namely, Path and Flatbed due to time limitation. List of considered attacks and attacks parameters are shown in 5.1.

Attack	Attack Values	Attack Position	Mode	Controller
PosInjectionAttack	[3, 5, 7, 9, 11] m	[0,2]	[Car-following,Middle Join,Exit]	Path, Flatbed
SpeedInjectionAttack	[-50, 0, 50, 100, 150] km/h	[0,2]	[Car-following,Middle Join,Exit]	Path, Flatbed
AccInjectionAttack	[-30, -10, 0, 10, 30] m/s2	[0,2]	[Car-following,Middle Join,Exit]	Path, Flatbed
GradualPosFalsificationAttack	[-10,40] m	[0,2]	[Car-following,Middle Join,Exit]	Path, Flatbed
GradualSpeedFalsificationAttack	[-10,17] m/s	[0,2]	[Car-following,Middle Join,Exit]	Path, Flatbed
GradualAccFalsificationAttack	[-10,10] m/s2 m	[0,2]	[Car-following,Middle Join,Exit]	Path, Flatbed
SmartPosFalsificationAttack	[-10,10] m	[0,2]	[Car-following,Middle Join,Exit]	Path, Flatbed
SmartSpeedFalsificationAttack	[-10,10] m/s	[0,2]	[Car-following,Middle Join,Exit]	Path, Flatbed
SmartAccFalsificationAttack	[-10,10] m/s2	[0,2]	[Car-following,Middle Join,Exit]	Path, Flatbed

Table 5.1: List of Attacks and Simulation Parameters [Source: [1]]

Table 5.2 summarizes the most relevant simulation parameters related to platoon configuration that controls the motion pattern generation. Leader speeds are applied for both controllers. Realistic sensors are used to reflect small inaccuracies in the simulations where small deviations randomly selected with uniform distribution. ϵ^{RAD} represents errors in radar sensors while ϵ^{V2V} represent errors in wireless communication. Warm up period is used to eliminate inaccuracies that can be induced by starting parameters. Default parameters are used for the presence of thermal noise. Delays in both physical and network layers are set to true.

Parameter	Values
Controller	PATH and Flatbed
Spacing	5m for both PATH an Flatbed controllers
Platoon Length	6 vehicles / 7 vehicles for middle join maneuver
Leader speed	50, 80, 100, 150 km/h
Sensor Uncertainties	$\epsilon_p^{V2V} = 1m$, $\epsilon_s^{V2V} = 0.1m/s$, $\epsilon_a^{V2V} = 0.01m/s^2$, $\epsilon_p^{RAD} = 0.1m$, $\epsilon_s^{RAD} = 0.01m/s$
Simulation duration	120s
Warm-Up Period	30s
Beacon frequency	10 Hz
Carrier Frequency	5.89 GHz
Max TX Power	100mW
Physical Layer Bitrate	6 Mbps
Sensitivity	-94dBm
Thermal Noise	-95dBm
Physical Layer Propagation Delay	True
Network Layer Propagation Delay	True

Table 5.2: Main Simulation Parameters [Source: [1]]

By considering listed twenty one attacks in table 5.1 for four speed scenarios and considering both PATH and Flatbed controllers during three maneuvers and two attacker positions, a total of thousand and eight tests have been executed. The method for platoon behavior recognition and anomaly detection has been

implemented using hmmlearn Python package which is a Python implementation of HMMs.

5.3 Results

In this section, experimental results from framework described in chapter 4 are presented. The first result that can be shown is the ability of proposed method to track the observations using a specified time window and determine the ongoing platoon behavior. It is an evaluation problem where a model which best matches the observations is chosen among several competing models representing each platoon behavior. Each platoon behavior has a label and the platoon behavior representing that label is printed out. As indicated earlier, number of components for each model representing a platoon behavior is learned using BIC criterion to select best model parameters for each Gaussian mixture. Number of components for car-following mode, middle join maneuver and exit maneuver are 10, 9 and 7 respectively. Models grammar for both platoon behavior recognition and anomaly detection are the same. After determining the ongoing platoon behavior and selecting the correct model, incoming observations with the same window size is used to detect anomalous observations. Window size of 10 samples which is equivalent to 1 sec is selected to enhance detection time. Detection results for each platoon behavior are presented in following subsections.

5.3.1 Platoon Behavior Recognition Performance

The main goal of the trained platoon behavior recognition component is to distinguish different maneuvers and normal lane following, named, car-following mode. The used signals are speed, acceleration and position. Results show that early recognition of the ongoing platoon behavior is possible with good accuracy. Evaluation data set consists of a separate driving sequences including 21 attack sequences for each platoon behavior considering two scenarios e.g., leader attack and follower attack.

Platoon behavior recognition is realized by splitting training data for each class representing a different platoon behavior. Each GMMHMM model representing the platoon behavior is trained with corresponding training data using Baum-Welch algorithm. Then, logarithmic likelihood of each model is calculated for the undergoing observations and model with maximum likelihood is chosen to recognize the platoon behavior. Log likelihood for both Path and Flatbed controllers are shown in Figure 5.1 and Figure 5.2 respectively.

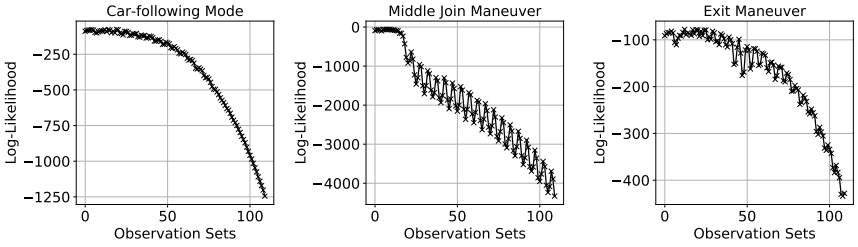


Figure 5.1: Log Likelihood of each Maneuver for Path Controller

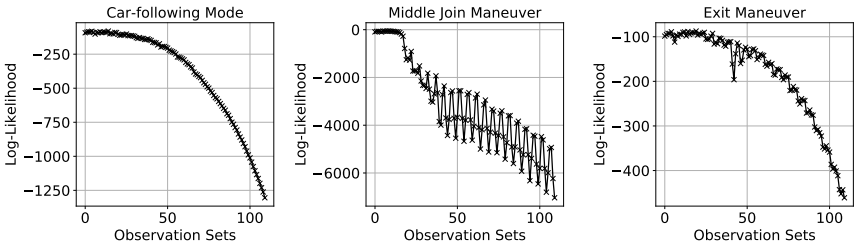


Figure 5.2: Log Likelihood of each Maneuver for Flatbed Controller

For car-following mode, middle join maneuver and exit maneuver, the best model was found to be a trained GMMHMM with seven, nine and seven components respectively. By plotting logarithmic likelihood, we can observe the different likelihood for each platoon behavior and the platoon behavior recognizer is expected to distinguish between different platoon behaviors.

As can be read from confusion matrix in Figure 5.3, the sensitivity of the platoon behavior recognition of all platoon behaviors are 100%.

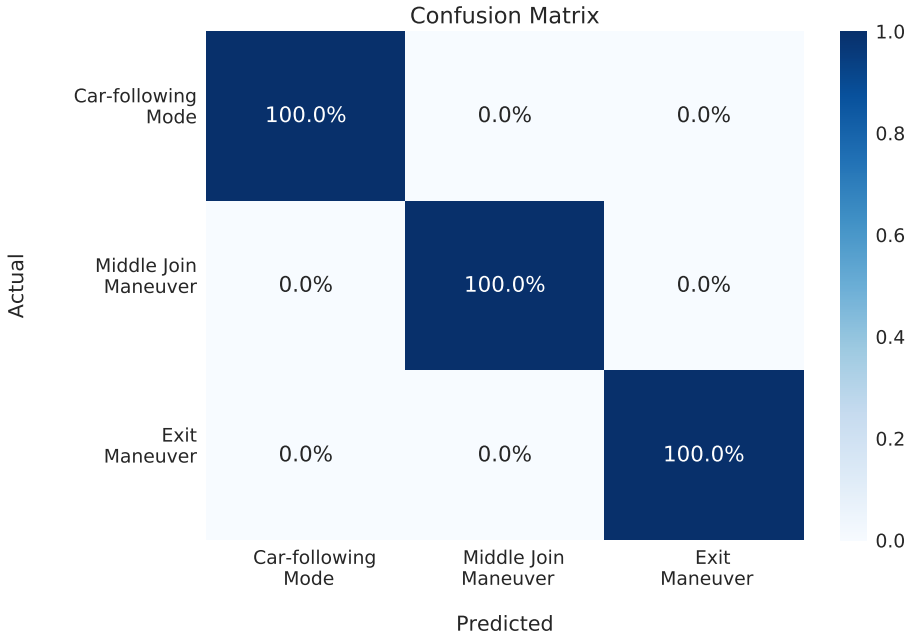


Figure 5.3: Confusion Matrix of Platoon Behavior Recognition

5.3.2 Anomaly Detection Performance

Results of different experiments carried out on normal and anomalous data are displayed in order to assess the performance of anomaly detection component. Only small set of tests are illustrated to help in drawing conclusions. Results for each platoon behavior are shown separately. Evaluation metrics explained in this chapter such as accuracy, precision, recall and F1 score are used to evaluate the performance of MDS. In addition, detection delay is used to evaluate time taken to detect misbehavior. Results are shown for each platoon behavior separately considering both controllers, Path and Flatbed.

Detection Delay

The proposed MDS works in a sliding window. Ten observation samples (1 second) are defined as a sliding window. Detection delay is calculated by adding processing delay (0.4s) and time between attack and end of sliding window. The results show that detection delay varies between 0.5s to 1.3s. It must be noted that if the attack occurred at the end of sliding window (i.e., last beacon), anomaly will be reported at the end of the sliding window which means after 0.1s. Accordingly, worst case scenario in terms of detection delay would happen if the attack is launched at the beginning of the sliding window as shown in Figure 5.4.

Comparing Performance of Anomaly Detection Approaches

Comparison between Viterbi-based and Forward-based anomaly detection is shown in Figure 5.5. AUC is used to compare the performance of these MDSs

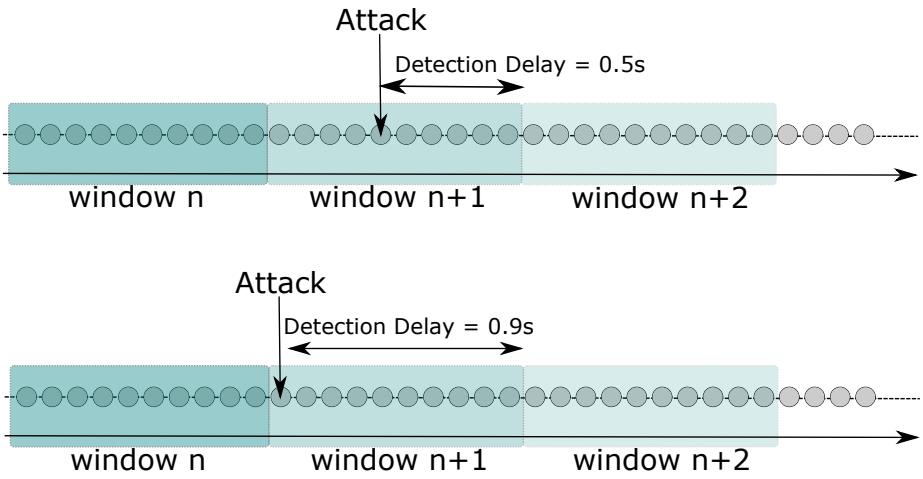
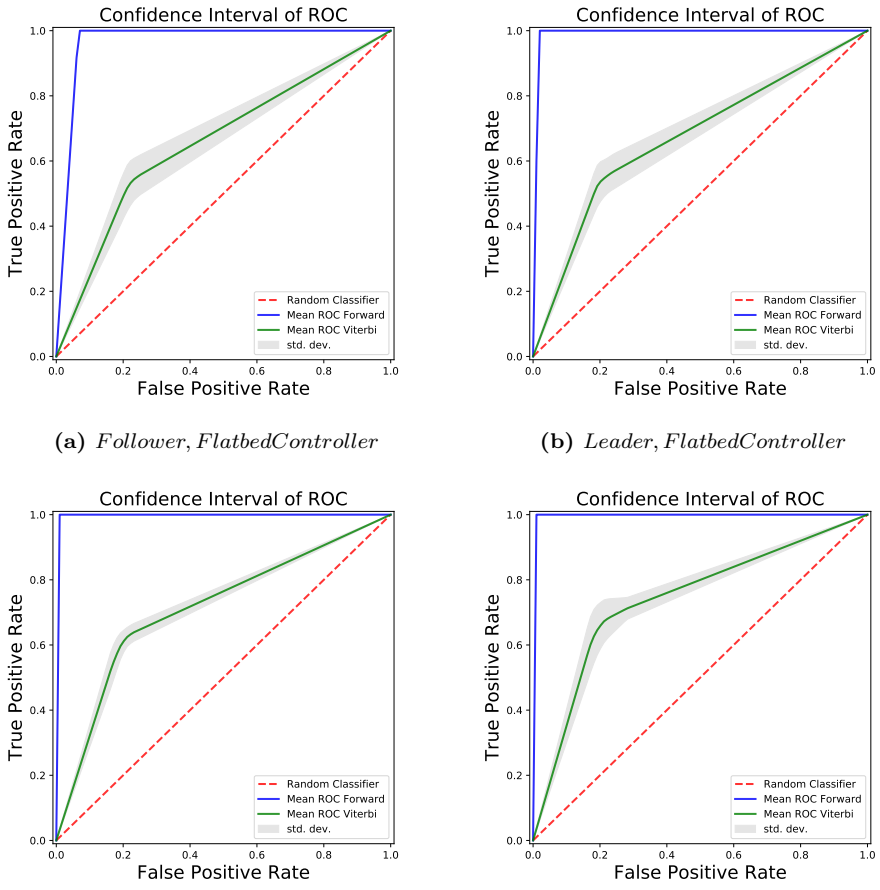


Figure 5.4: Detection Delay

by running the experiment 50 times for one attack. Figures 5.5(a) and 5.5(b) show the performance comparison for Flatbed controller for different attacker positions. It can be seen that Forward-based algorithm is approaching 1 which means that the model has a good performance in separating the positive class (e.g., misbehavior) and the negative class (e.g., benign behavior). On the other hand, Viterbi-based algorithm shows lower model performance with AUC less than 0.6. Performance for Path controller is shown in Figures 5.5(c) and 5.5(d). In these figures, more acceptable performance is witnessed for Viterbi-based algorithm with AUC more than 0.6 and may reach 0.7. However, Forward-based algorithm shows a consistent performance reaching 1 similar to Flatbed controller results. For both controllers, the performance show that for small False Positive Rate (FPR) values, the AUC curves for Viterbi-based algorithm and Forward-based algorithm exhibit essential differences. Based on these results, Forward-based approach is adopted and its performance is shown solely in the following sections.



(a) *Follower*, *FlatbedController*

(b) *Leader*, *FlatbedController*

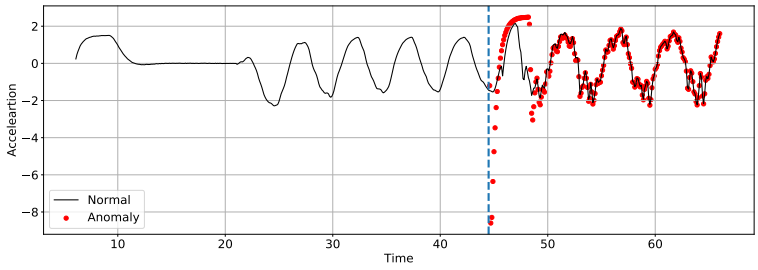
(c) *Follower*, *PathController*

(d) *Leader*, *PathController*

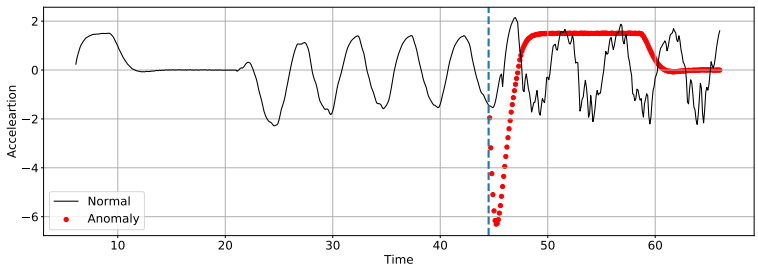
Figure 5.5: ROC Curve for Viterbi-based Algorithm and Forward-based Algorithm (Acceleration Injection Attack at Speed = 50)

Anomaly Detection During Middle Join Maneuver

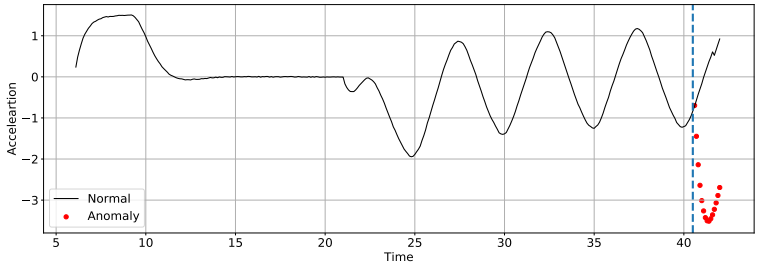
Results for middle join maneuver are shown in this section. Figure 5.6 shows detected anomalies of smart acceleration falsification attack when the platoon drive with speed 50 km/h. Black line represents the normal behavior of acceleration of attacked vehicle while red dots represent detected anomalies by the MDS. Vertical blue line indicates the time of attack as configured in the simulations. A common observation is the accurate detection of the attack. It can be seen also that the impact is clearer when the attack is conducted by leader and accordingly, it can be easily detected by MDS. When the attack is conducted by follower vehicle, the attack can be detected but it can be noticed that detected anomalies are close to normal behavior for flatbed controller. This was a driver for adopting sensitive MDS.



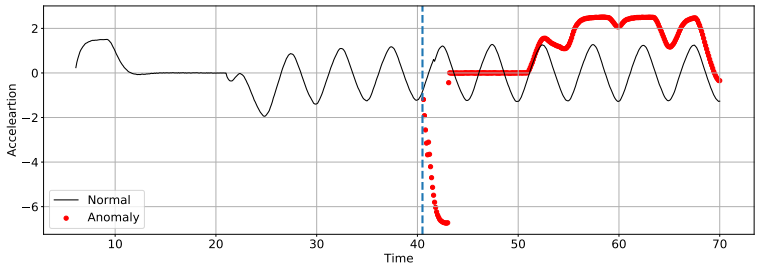
(a) *Follower*, *FlatbedController*



(b) *Leader*, *FlatbedController*



(c) *Follower*, *PathController*

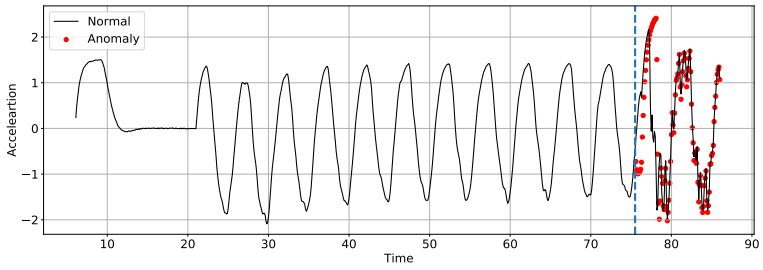


(d) *Leader*, *PathController*

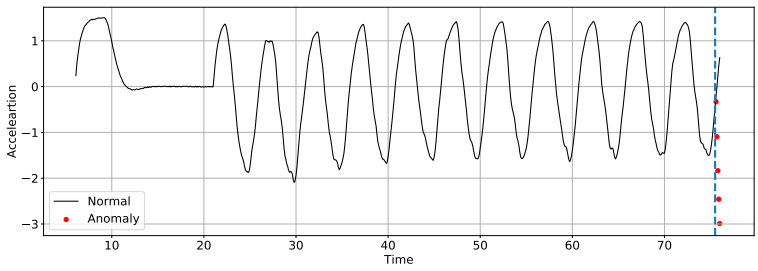
Figure 5.6: Smart Acceleration Falsification Attack During Middle Join Maneuver at Speed = 50

Figure 5.7 shows detected anomalies of gradual speed falsification attack when the platoon drive with speed 80 km/h. It can be seen that impact on both

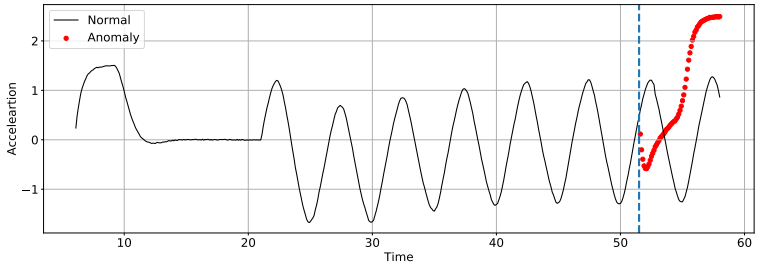
controllers varies from causing a crash to destabilizing the platoon. Figure 5.7(b) indicate that a crash happened and simulation stopped. This shows the value of ability to detect anomaly as early as possible and react to it before crash. On the other hand, Figure 5.7(a) shows that if attack is conducted by a follower, it has less impact since the controller relies on speed information received by leader. Impact of this attack on PATH is high in terms of platoon destabilization as shown in Figure 5.7(c) and Figure 5.7(d). MDS was able to detect all deviations resulted from this attack accurately.



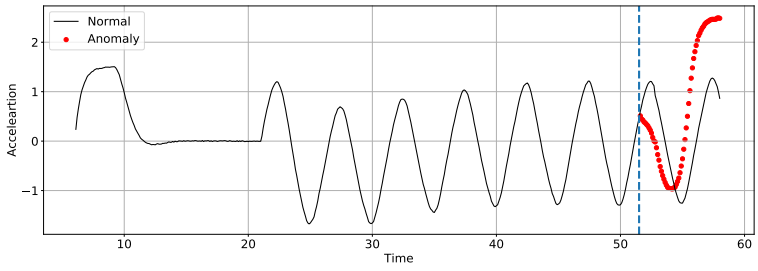
(a) *Follower, FlatbedController*



(b) *Leader, FlatbedController*



(c) *Follower, PathController*

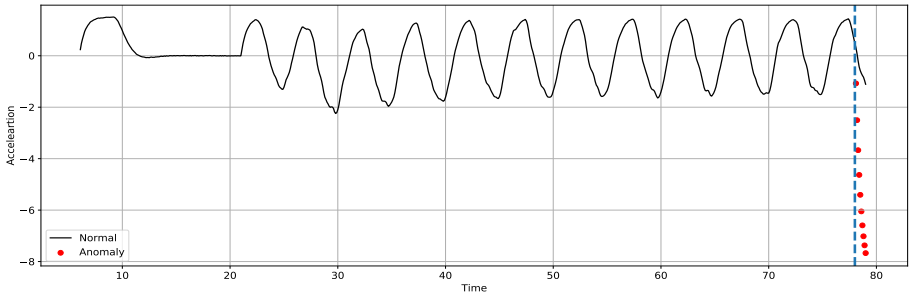


(d) *Leader, PathController*

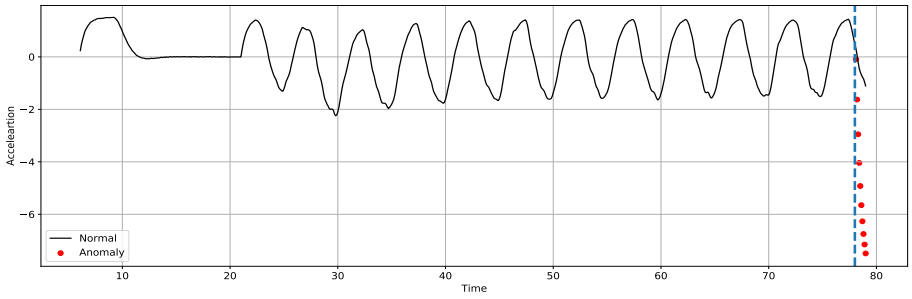
Figure 5.7: Gradual Speed Falsification Attack During Middle Join Maneuver at Speed = 80

Similar results are shown in Figure 5.8 for speed injection attack with platoon speed of 100 km/h. PATH results are similar with high divergence from nominal

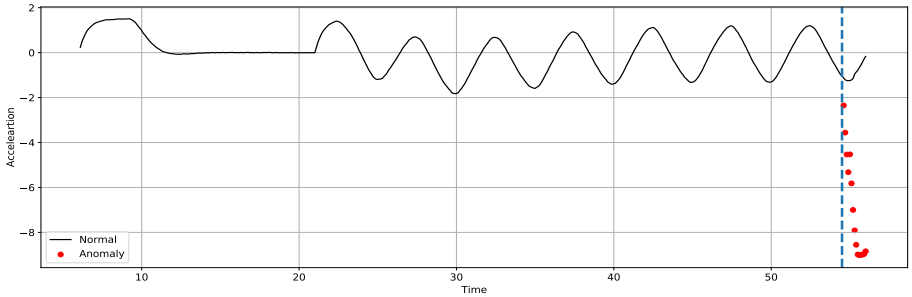
behavior whether the attack is initiated from platoon leader or a follower in the platoon as illustrated in Figure 5.8(c) and Figure 5.8(d). Similarly, Flatbed controller is impacted and result in crash with high speed in malicious leader scenario while impact is less if the attack is conducted by follower vehicle.



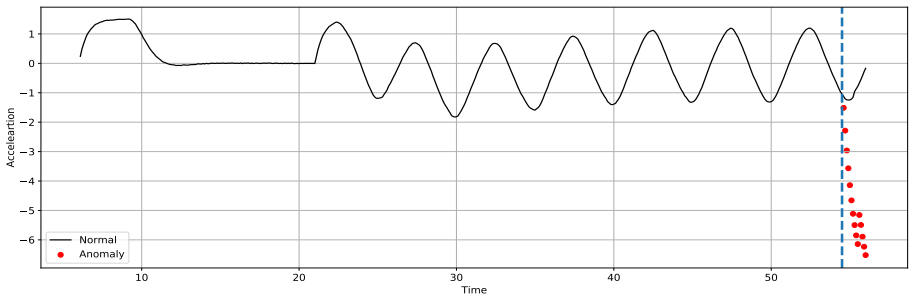
(a) *Follower, FlatbedController*



(b) *Leader, FlatbedController*



(c) *Follower, PathController*

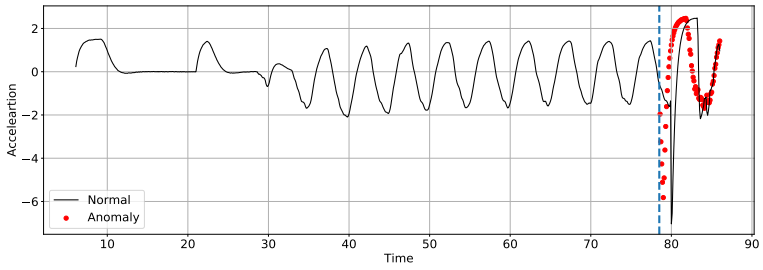


(d) *Leader, PathController*

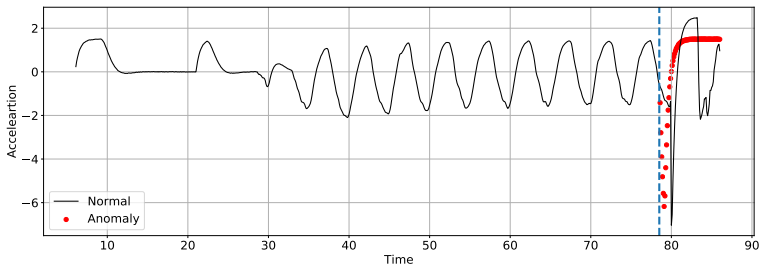
Figure 5.8: Speed Injection Attack During Middle Join Maneuver at Speed = 100

Figure 5.9 shows smart position falsification attack with platoon speed of 150 km/h. it is an interesting attack because both Flatbed and PATH are resilient to position attacks since position information are not used as an input to these

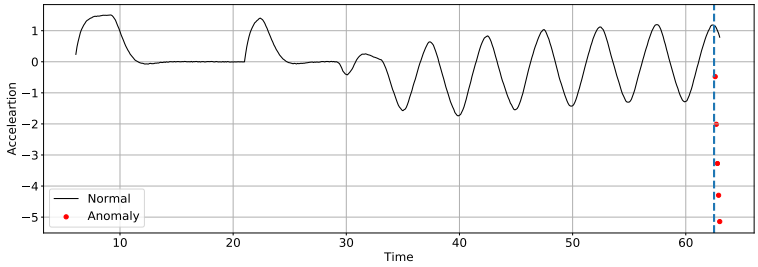
controllers. However, since this attack changes other variables (e.g., speed and acceleration) according to kinematic model, impact can be seen as more severe and actually results in a crash for both controllers. PATH controller also resulted in a severe crash as shown in Figure 5.9(d).



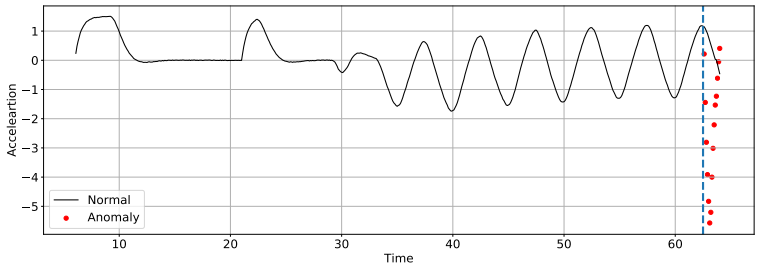
(a) *Follower*, *FlatbedController*



(b) *Leader*, *FlatbedController*



(c) *Follower*, *PathController*

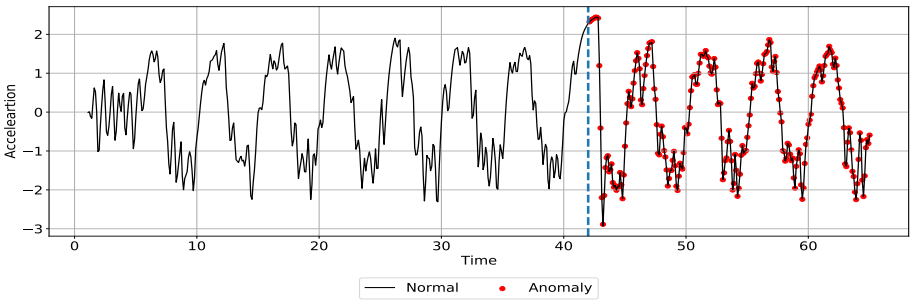


(d) *Leader*, *PathController*

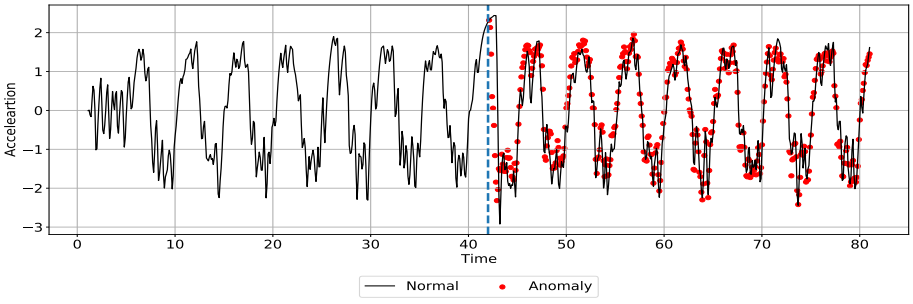
Figure 5.9: Smart Position Falsification Attack During Middle Join Maneuver at Speed = 150

Anomaly Detection During Exit Maneuver

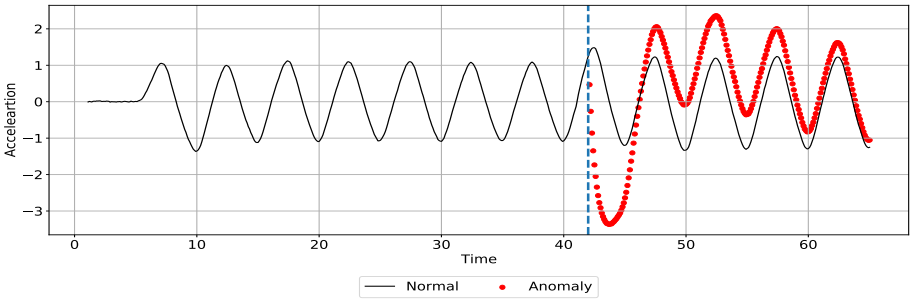
Results for exit maneuver are shown in this section. Figure 5.10 shows detected anomalies of smart acceleration falsification attack when the platoon drive with speed 50 km/h. As illustrated in Figure 5.10(a) and Figure 5.10(b), impact on Flatbed controller is different comparing to middle join maneuver. Slight deviation can be seen for both malicious follower and malicious leader scenarios. Sensitive MDS help in such scenario since it was able to detect minor deviations from nominal behavior. Although the impact would not be severe, it is important to detect that a malicious actor is part of platoon, report him and evict him from platooning formation to prevent him from conducting other attacks that could have more severe impact.



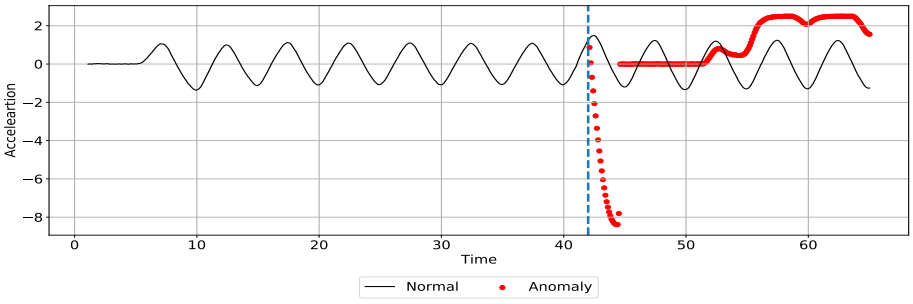
(a) *Follower, FlatbedController*



(b) *Leader, FlatbedController*



(c) *Follower, PathController*

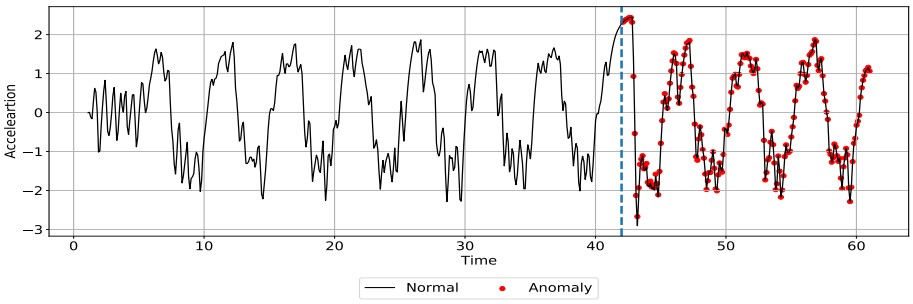


(d) *Leader, PathController*

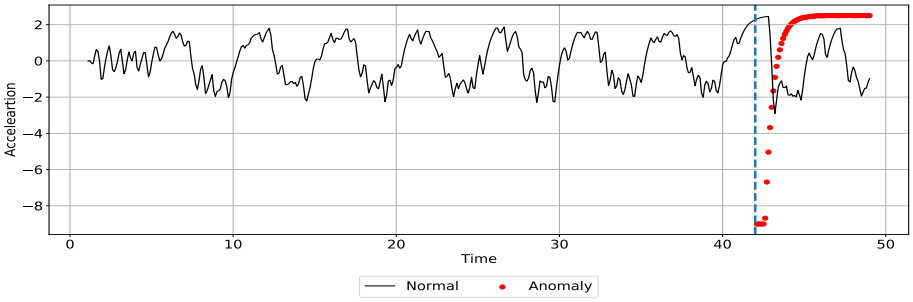
Figure 5.10: Smart Acceleration Falsification Attack During Exit Maneuver at Speed = 50

Gradual speed attack during exit maneuver with speed 80 km/h is shown in Figure 5.11. Impact on Flatbed controller is similar to impact during middle join

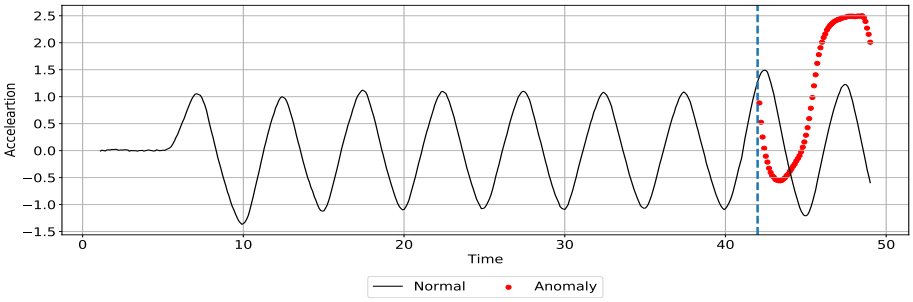
maneuver when initiated by leader as shown in Figure 5.11(b). However, when attack is initiated by follower, impact is less in terms of severity when compared with middle join as shown in Figure 5.11(a). Nevertheless, small deviations are detected accurately and correctly. On the other hand, impact on PATH controller is similar to impact during middle join.



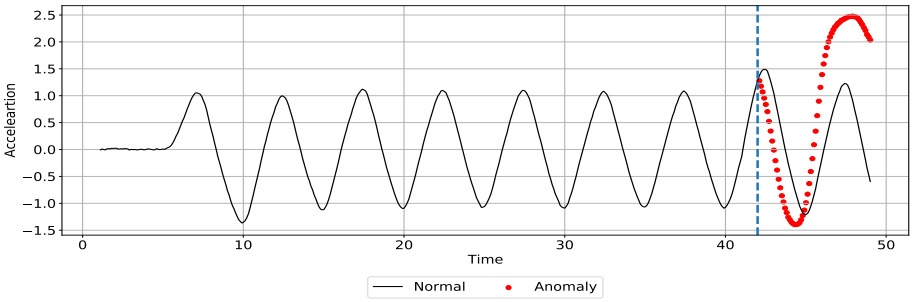
(a) *Follower, FlatbedController*



(b) *Leader, FlatbedController*



(c) *Follower, PathController*

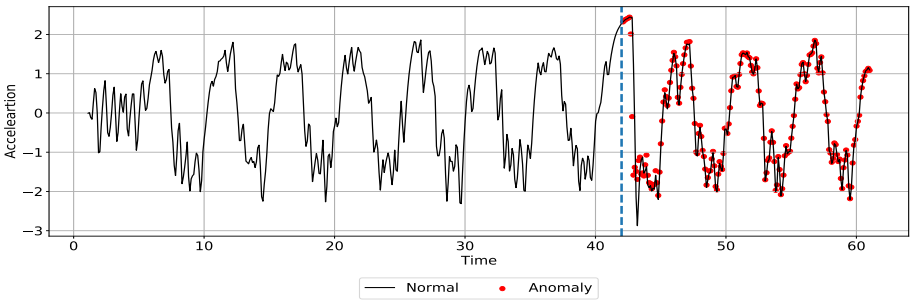


(d) *Leader, PathController*

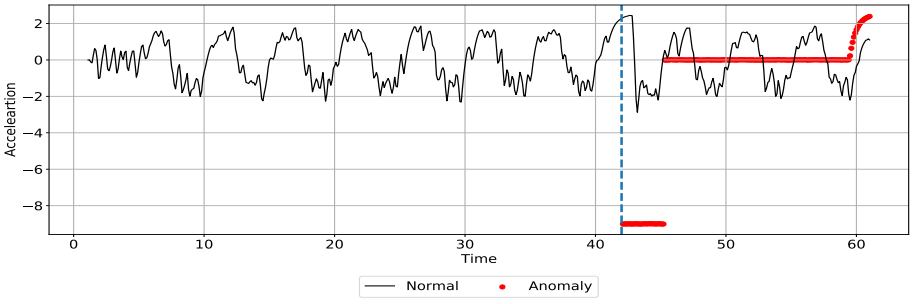
Figure 5.11: Gradual Speed Falsification Attack During Exit Maneuver at Speed = 80

Results for speed injection attack with platoon speed of 100 km/h are shown in Figure 5.12. Figure 5.12(a) and Figure 5.12(b) show different impact on Flatbed

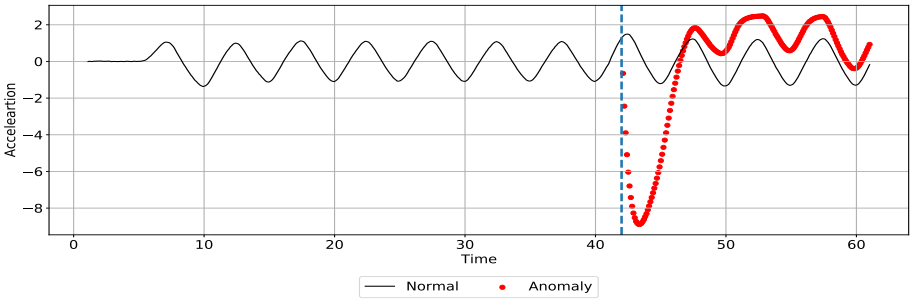
controller when compared with middle join with slight impact in malicious follower scenario while more severe impact in malicious leader. Path controller is also impacted similar to join maneuver but it is not the exact impact as shown in Figure 5.12(c) and Figure 5.12(d).



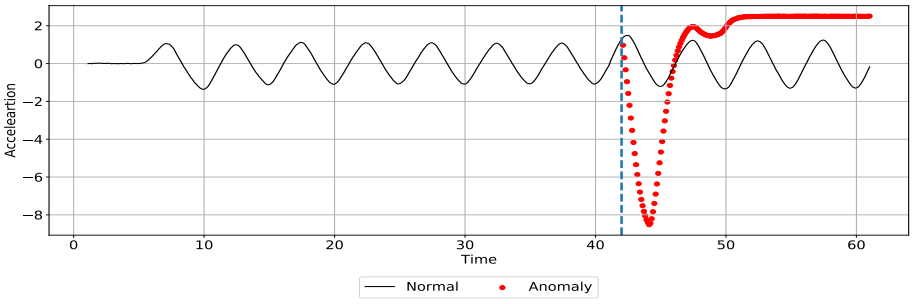
(a) *Follower, FlatbedController*



(b) *Leader, FlatbedController*



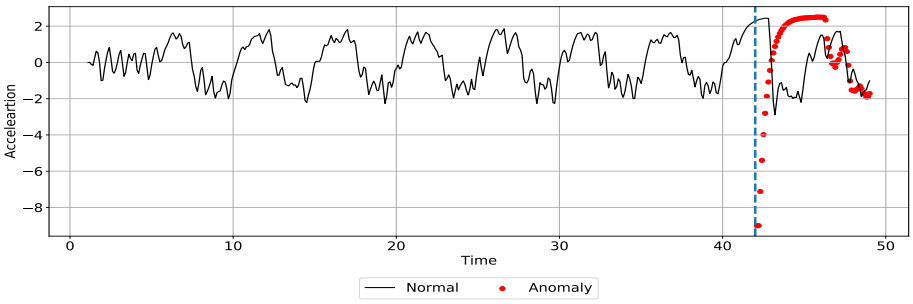
(c) *Follower, PathController*



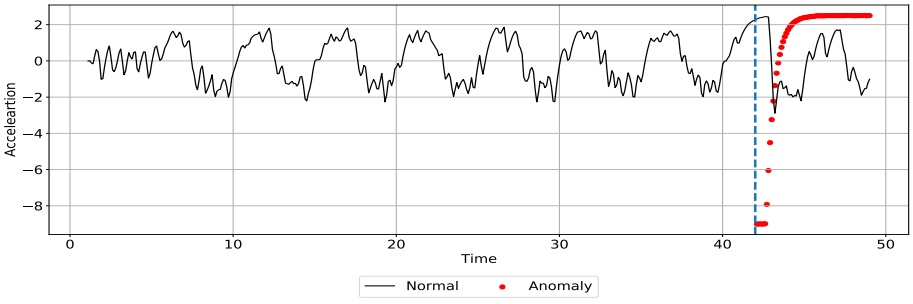
(d) *Leader, PathController*

Figure 5.12: Speed Injection Attack During Exit Maneuver at Speed = 100

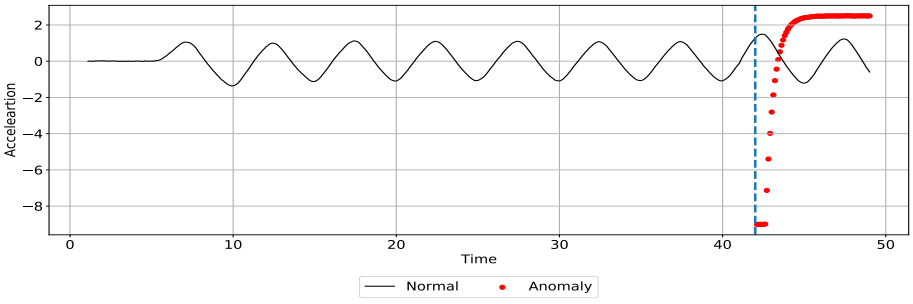
According to Figure 5.13, impact of smart position falsification attack during exit maneuver is almost similar to impact of same attack during middle join maneuver. All position falsification attempts have been accurately spotted.



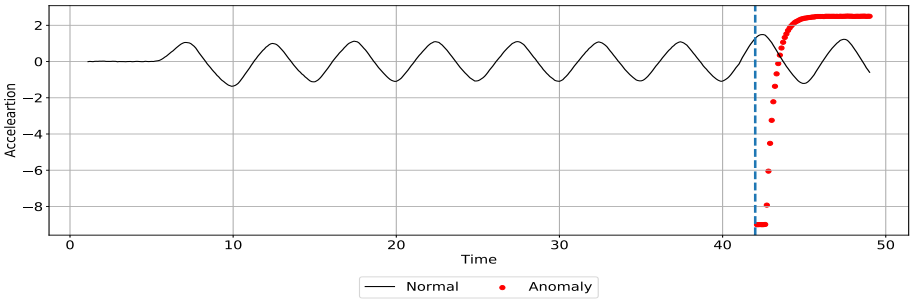
(a) *Follower, FlatbedController*



(b) *Leader, FlatbedController*



(c) *Follower, PathController*

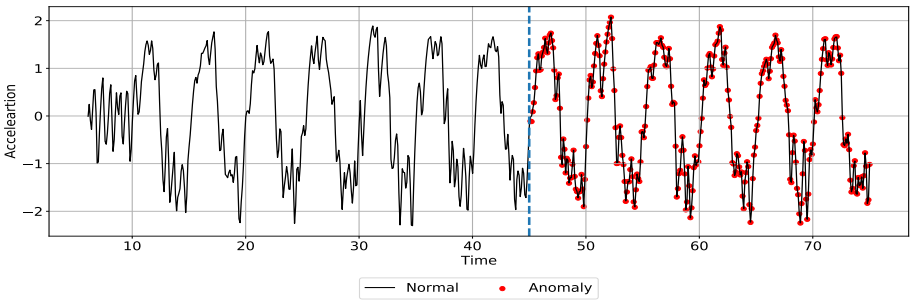


(d) *Leader, PathController*

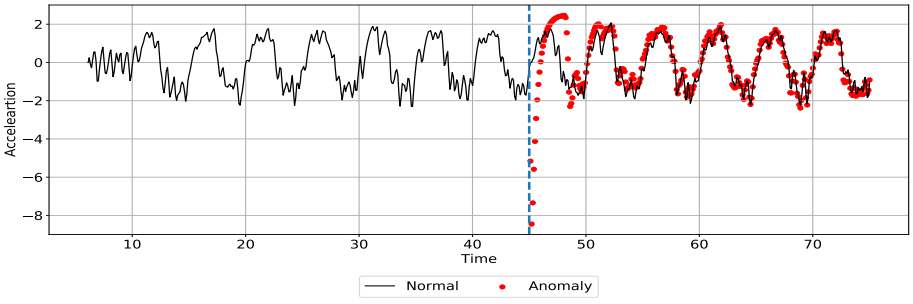
Figure 5.13: Smart Position Falsification Attack During Exit Maneuver at Speed = 150

Anomaly Detection During Car-following Mode

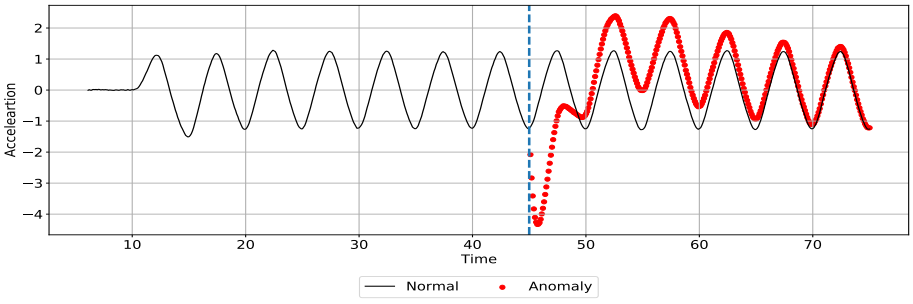
Results of performance of MDS for smart position falsification attack with platoon speed of 50 km/h are shown in Figure 5.14. Slight impact on Flatbed controller during car-following mode similar to exit maneuver as shown in Figure 5.14(a) and Figure 5.14(b). MDS was able to detect this slight deviation accurately. Impact on PATH controller is almost similar to impact during middle join and exit maneuvers. Deviation was obvious and MDS was able to detect the misbehavior easily as shown in Figure 5.14(c) and Figure 5.14(d).



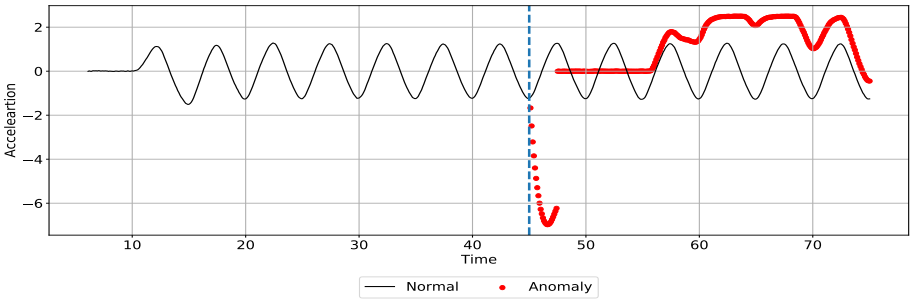
(a) *Follower*, *FlatbedController*



(b) *Leader*, *FlatbedController*



(c) *Follower*, *PathController*

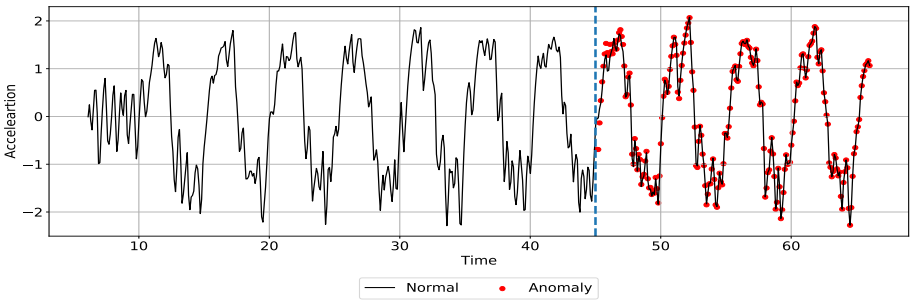


(d) *Leader*, *PathController*

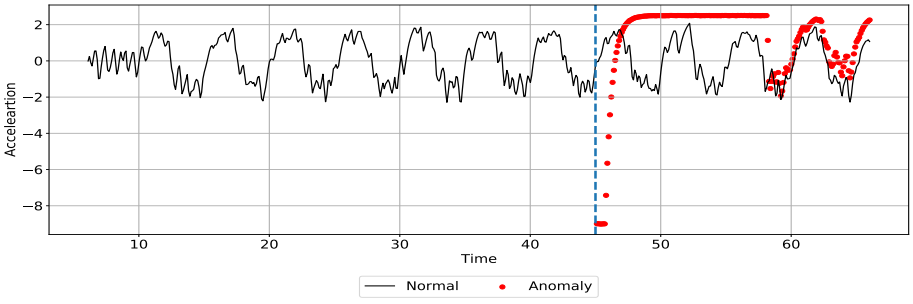
Figure 5.14: Smart Acceleration Falsification Attack During Car-following Mode at Speed = 50

Results of impact and detection for smart acceleration falsification attack during car-following mode when the platoon is moving with speed 80 km/h is

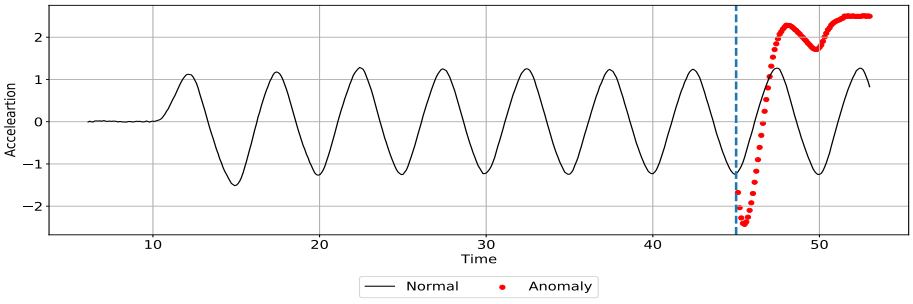
shown in Figure 5.15. For Flatbed controller, slight impact on behavior is shown in Figure 5.15(a) and Figure 5.15(b) similar to middle join and exit maneuvers whether the attack is initiated by follower or leader. For PATH controller, small deviation is indicated at the beginning of the attack then the deviation become more clearer and easier to detect as shown in Figure 5.15(c) and Figure 5.15(d).



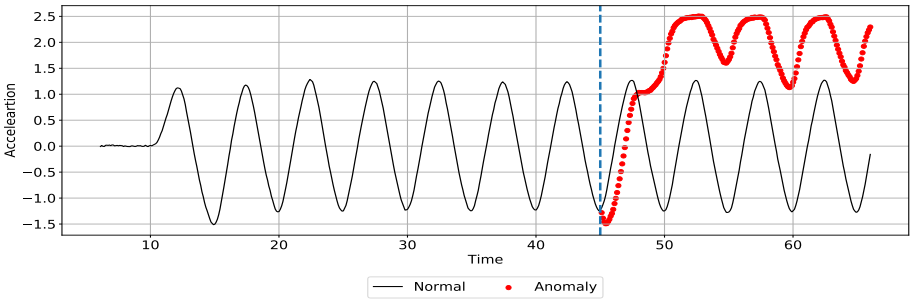
(a) *Follower, FlatbedController*



(b) *Leader, FlatbedController*



(c) *Follower, PathController*

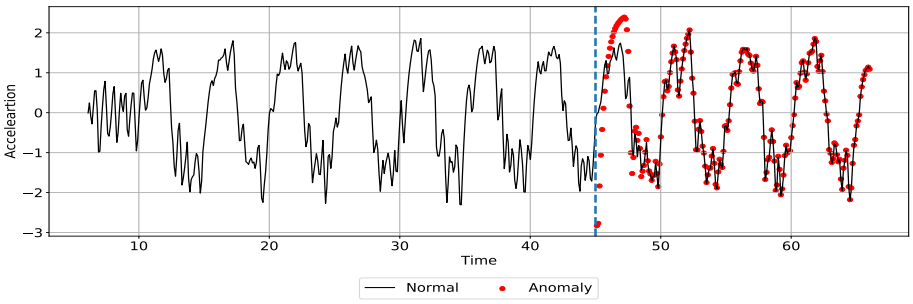


(d) *Leader, PathController*

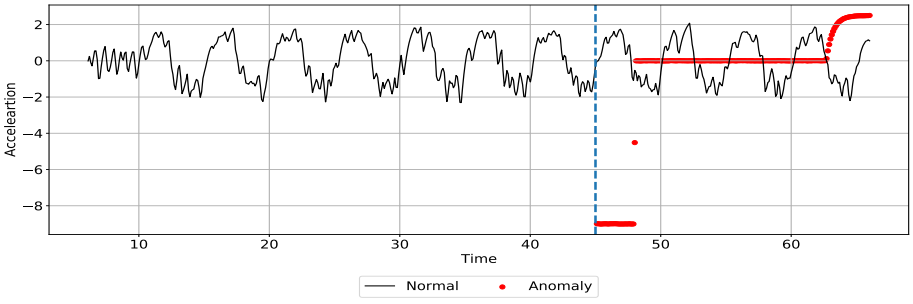
Figure 5.15: Gradual Speed Falsification Attack During Car-following Mode at Speed = 80

MDS performance under speed injection attack during car-following mode with platoon driving at 100 km/h is shown in Figure 5.16. Impact is almost

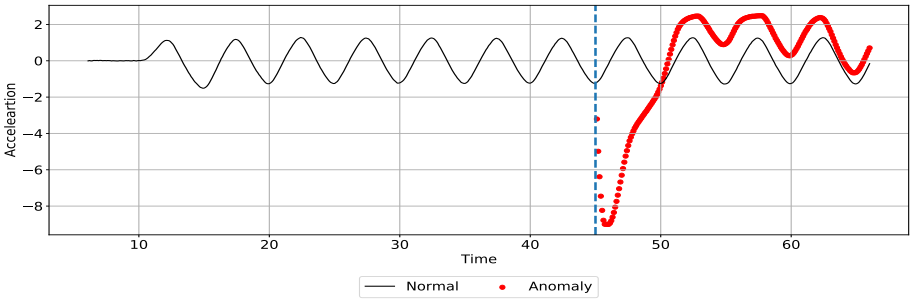
similar to exit maneuver. It is also similar to impact during middle join maneuver for PATH controller as shown in Figure 5.16(c) and Figure 5.16(d). On the other hand, it differs for Flatbed controller when compared with middle join maneuver as shown in Figure 5.16(a) and Figure 5.16(b).



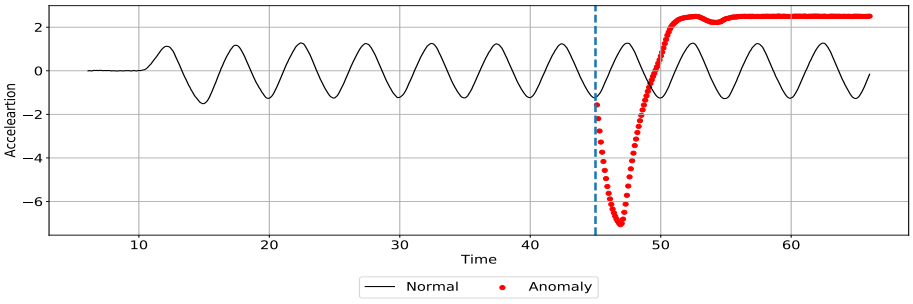
(a) *Follower, FlatbedController*



(b) *Leader, FlatbedController*



(c) *Follower, PathController*

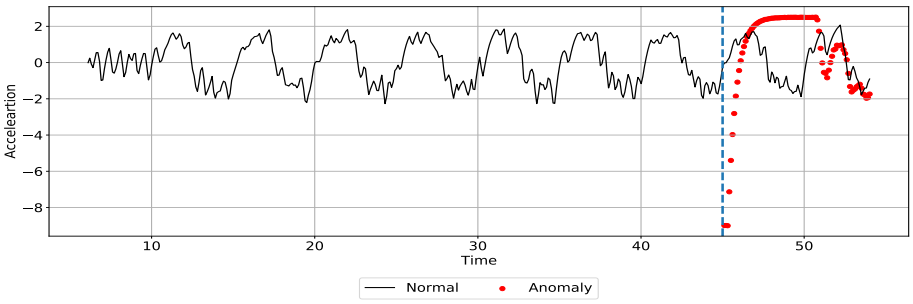


(d) *Leader, PathController*

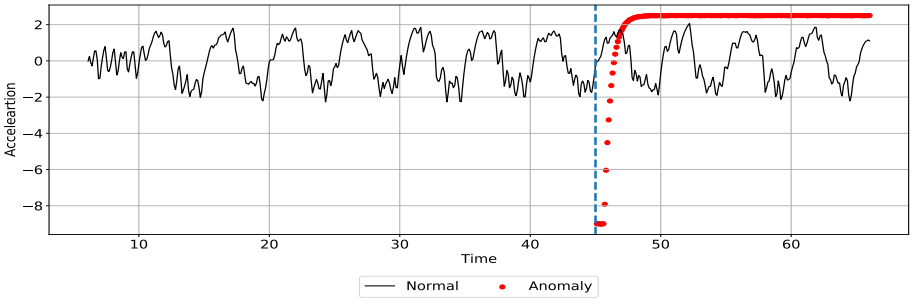
Figure 5.16: Speed Injection Attack During Car-following Mode at Speed = 100

Similarly, MDS performance under smart position falsification attack during car-following mode with platoon driving at 150 km/h is shown in Figure 5.17. Impact on both Flatbed and PATH controllers is similar to exit maneuver.

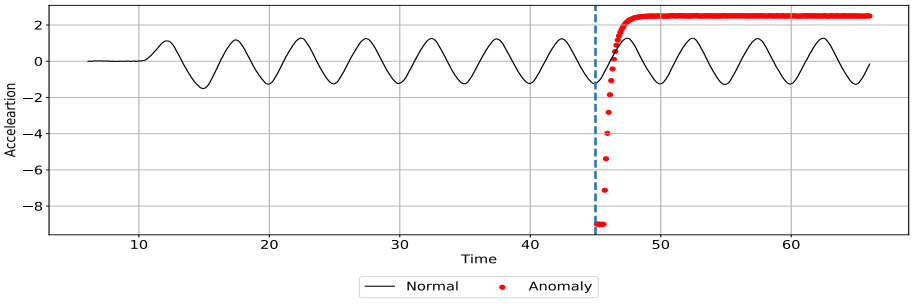
Impact on PATH and Flatbed is also similar to middle join maneuver if the attack is initiated by leader as shown in Figure 5.17(b) and Figure 5.17(d). However, difference can be seen in impact when compared with middle join if the attack is initiated by follower as shown in Figure 5.17(a) and Figure 5.17(c).



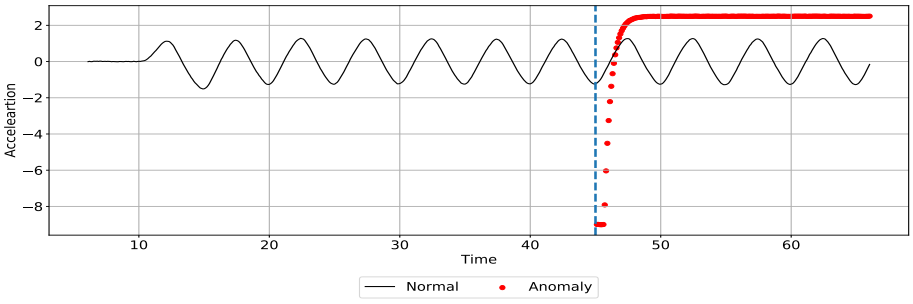
(a) *Follower, FlatbedController*



(b) *Leader, FlatbedController*



(c) *Follower, PathController*



(d) *Leader, PathController*

Figure 5.17: Smart Position Falsification Attack During Car-following Mode at Speed = 150

Chapter 6

Conclusion and Future Work

6.1 Conclusion

In this thesis, nominal behavior of platoon behavior is investigated in order to have a better understanding of factors that need to be considered in modelling this behavior. It has been noticed that three dimensions need to be considered. These dimensions are the undergoing platoon behavior itself, the speed during the platoon behavior and the platooning controller. These aspects will affect the distribution of data and need to be considered while modeling the nominal behavior and accordingly the design of local MDS in each vehicle. Since local MDS must be aware of undergoing platoon behavior, platoon behavior recognition component is added to the framework to recognize the undergoing platoon behavior and intelligently select the model representing the nominal behavior. The selected model will be used to detect any deviation from nominal behavior. platoon behavior recognition was able to recognize a set of driving platoon behaviors at an early stage even if only a limited set of input signals are used (small window of 1 second). The performance of platoon behavior recognition showed excellent performance by predicting undergoing platoon behavior correctly. Different approaches can be used for anomaly detection. Two approached are covered in this thesis to detect twenty one different attacks. One approach is to use Viterbi algorithm to decode and compute the hidden states corresponding to observation sequence for both nominal behavior and attack and compare the to detect differences and report them as anomalies. Another approach is to use forward-backward algorithm to estimate the probability/likelihood of observation sequence. Second approach showed better and consistent performance in terms of performance metrics such as accuracy, precision, recall and f1 score. It has been shown that a single run is enough for learning the normal behavior. The MDS is very sensitive and accordingly they are able to detect very small deviations. This high sensitivity can introduce false alarms for jerkiness scenarios.

6.2 Future Work

Proposed new framework showed promising results in detecting misbehavior during platooning management. However, some important limitations apply for this framework which can be addressed by future work.

6.2.1 Used ML Features Challenges

Used ML features are obvious to use since they are reflecting kinematic model. However, this choice raise the challenge of the need for building a model for each speed. An improvement in this area is to re-engineer the ML features in a way that will contribute to the to achieve better generalization of the ML model without impacting the performance of the MDS.

6.2.2 Misbehavior Reports Sharing Challenges

It is expected that a local MDS will be deployed in each vehicle. More investigations needed by running the framework on all vehicles and compare results to see if sharing misbehavior reports between vehicles will make the the framework more robust. Sharing misbehavior reports with other vehicles opens the issue of fake reports. Cooperative schemes assume that reports from other vehicles are trustworthy if they carry correct signatures. Validation of received misbehavior reports need to be investigated.

6.2.3 Meeting 3GPP Requirements for Platooning Application

3GPP put tight end to end latency requirements for platooning reaching 10 ms. Accordingly, reducing sliding window for both platoon behavior recognition and anomaly detection without impacting the performance need to be investigated.

6.2.4 Deployment in Vehicles and Real Time Performance

A local MDS shall be deployed in vehicles and accordingly it needs to consider real time characteristics and feed of information in streaming fashion. In addition, hardware requirements need to be considered. Other aspects such as secure storage of ML models in ML model repository which will be located in each vehicles need to be investigated. Other aspect need to be considered is best location to deploy MDS. MDS can be deployed before controller and accordingly detect all attackers and have the ability exclude the attacker from platooning formation. However, disturbances that could occur due to driving over hilly terrain will be treated as misbehavior. On the other hand, if an MDS is deployed after the controller, these changes will be accommodated by the controller and MDS will not detect these deviations as misbehavior. However, if the controller is resilient to a certain attacks, these attacks will not be captured by a local MDS which will raise the risk of having attacker as a part of the platoon for longer time. Accordingly, these options need to be investigated further.

6.2.5 New Approaches to Measure Dissimilarity with Normal Behavior

Any deviation from normal behavior was considered as an anomaly and accordingly, the framework was able to detect slight deviations from nominal behavior. However, this sensitivity can result in having false positives in case of jerky behavior (i.e., sudden acceleration changes). Although it is considered as a safety risk, this behavior can be conducted by a vehicle outside the platoon but the result of this jerk behavior will impact the platoon. This would result in detecting this behavior as an attack and accordingly, dissolving the platoon unnecessarily. Some techniques are investigated in the context of the thesis but not thoroughly due to time limitation such as Hellinger distance, Kullback Leibler Divergence and other techniques. These approaches can be investigated further in future work.

Bibliography

[1] Konstantinos Kalogiannis. Investigating Attacks on Vehicular Platooning and Cooperative Adaptive Cruise Control. Master's thesis, Royal Institute of Technology (KTH), Stockholm, Sweden, December 2020.

[2] Panos Papadimitratos, A. de La Fortelle, K. Evenssen, R. Brignolo, and S. Cosenza. Vehicular Communication Systems: Enabling Technologies, Applications, and Future Outlook on Intelligent Transportation. *IEEE Communications Magazine*, 47(11):84–95, November 2009.

[3] ETSI TS 102 637-2, Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 2: Specification of Co-operative Awareness Basic Service. March 2011.

[4] ETSI EN 302 637-3 V1.2.1. Intelligent Transport Systems (ITS); Vehicular Communications; Basic Set of Applications; Part 3: Specifications of Decentralized Environmental Notification Basic Service. September 2014.

[5] SAE International. Dedicated Short Range Communications (DSRC) Message Set Dictionary. Technical report J2735. November 2009.

[6] Robert Mitchell and Ing-Ray Chen. A Survey of Intrusion Detection Techniques for Cyber-Physical Systems. *ACM Comput. Surv.*, 46(4), March 2014.

[7] Mani Amoozadeh, Hui Deng, Chen-Nee Chuah, H Michael Zhang, and Dipak Ghosal. Platoon Management with Cooperative Adaptive Cruise Control Enabled by VANET. *Veh. Comm.*, 2(2):110–123, April 2015.

[8] Dongyao Jia, Kejie Lu, Jianping Wang, Xiang Zhang, and Xuemin Shen. A Survey on Platoon-Based Vehicular Cyber-Physical Systems. *IEEE Communications Surveys & Tutorials*, 18(1):263–284, March 2015.

[9] ETSI TR 103 298 V0.0.4. Intelligent Transport Systems (ITS); Platooning; Pre-standardization study. January 2019.

[10] Zeinab El-Rewini, Karthikeyan Sadatsharan, Niroop Sugunaraj, Daisy Flora Selvaraj, Siby Jose Plathottam, and Prakash Ranganathan. Cybersecurity challenges in vehicular communications. *Veh. Comm.*, 23:100214, June 2020.

[11] Lijian Xu, Le Yi Wang, George Yin, and Hongwei Zhang. Communication information structures and contents for enhanced safety of highway vehicle platoons. *IEEE Transactions on Vehicular Technology*, 63(9):4206–4220, November 2014.

[12] Tim Leinmuller, Elmar Schoch, and Christian Maihofer. Security requirements and solution concepts in vehicular ad hoc networks. pages 84–91, January 2007.

[13] S. Gisdakis, M. LaganÃ, T. Giannetsos, and P. Papadimitratos. SEROSA: Service Oriented Security Architecture for Vehicular Communications. In *IEEE Vehicular Networking Conference (IEEE VNC)*, pages 111–118, Boston, MA, USA, December 2013.

[14] N. Alexiou, S. Gisdakis, M. LaganÃ, and P. Papadimitratos. Towards a Secure and Privacy-preserving Multi-service Vehicular Architecture. In *IEEE Workshop on Data Security and Privacy in Wireless Networks (IEEE D-SPAN), collocated with IEEE WoWMoM*, pages 1–6, Madrid, Spain, June 2013.

[15] Mohammad Khodaei, Hongyu Jin, and Panos Papadimitratos. Towards Deploying a Scalable & Robust Vehicular Identity and Credential Management Infrastructure. In *IEEE Vehicular Networking Conference (IEEE VNC)*, pages 33–40, Paderborn, Germany, December 2014.

[16] Nikolaos Alexiou, Marcello Laganà, Stylianos Gisdakis, Mohammad Khodaei, and Panagiotis Papadimitratos. VeSPA: Vehicular Security and Privacy-preserving Architecture. In *ACM Workshop on Hot Topics on Wireless Network Security and Privacy (ACM HotWiSec)*, pages 19–24, Budapest, Hungary, April 2013.

[17] Mohammad Khodaei, Hamid Noroozi, and Panos Papadimitratos. Scaling Pseudonymous Authentication for Large Mobile Systems. In *Proceedings of the 12th ACM Conference on Security & Privacy in Wireless and Mobile Networks (ACM WiSec)*, pages 174–185, Miami, FL, USA, May 2019.

[18] M. Khodaei, H. Jin, and P. Papadimitratos. SECMACE: Scalable and Robust Identity and Credential Management Infrastructure in Vehicular Communication Systems. *IEEE Transactions on Intelligent Transportation Systems*, 19(5):1430–1444, May 2018.

[19] W. Whyte, A Weimerskirch, V. Kumar, and T. Hehn. A Security Credential Management System for V2V Communications. In *IEEE Vehicular Networking Conference (VNC)*, Boston, MA, December 2013.

[20] Maxim Raya, Panos Papadimitratos, Virgil Gligor, and Jean-Pierre Hubaux. On Data-Centric Trust Establishment in Ephemeral Ad hoc Networks. In *IEEE Conference on Computer Communications (IEEE INFOCOM)*, pages 1238–1246, Phoenix, AZ, USA, April 2008.

[21] Maxim Raya, Panos Papadimitratos, Imad Aad, Dan Jungels, and Jean-Pierre Hubaux. Eviction of Misbehaving and Faulty Nodes in Vehicular Networks. *IEEE Journal on Selected Areas in Communications (IEEE JSAC)*, 25(8):1557–1568, October 2007.

[22] R. W. van der Heijden, S. Dietzel, T. Leinmüller, and F. Kargl. Survey on misbehavior detection in cooperative intelligent transportation systems. *IEEE Communications Surveys & Tutorials*, 21(1):779–811, October 2018.

[23] Keno Garlichs, Alexander Willecke, Martin Wegner, and Lars C Wolf. TriP: Misbehavior Detection for Dynamic Platoons using Trust. In *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, pages 455–460, Auckland, New Zealand, October 2019. IEEE.

[24] Faris Alotibi and Mai Abdelhakim. 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, April 2020.

[25] Marco Iorio, Fulvio Risso, Riccardo Sisto, Alberto Buttiglieri, and Massimo Reineri. Detecting injection attacks on cooperative adaptive cruise control. In *2019 IEEE Vehicular Networking Conference (VNC)*, pages 1–8, Los Angeles, CA, USA, December 2019. IEEE.

[26] Bruce DeBruhl, Sean Weerakkody, Bruno Sinopoli, and Patrick Tague. Is your commute driving you crazy? A study of misbehavior in vehicular platoons. In *Proceedings of the 8th ACM Conference on Security & Privacy in Wireless and Mobile Networks*, pages 1–11, New York, NY, USA, June 2015.

[27] Panos Papadimitratos, V. Gligor, and Jean-Pierre Hubaux. Securing Vehicular Communications-Assumptions, Requirements, and Principles. In *Workshop on Embedded Security in Cars (ESCAR)*, Berlin, Germany, November 2006.

[28] R. Rajamani. Vehicle Dynamics and Control. December 2011.

[29] Christopher Nowakowski, Steven E Shladover, Xiao-Yun Lu, Deborah Thompson, and Aravind Kailas. Cooperative adaptive cruise control (cacc) for truck platooning: Operational concept alternatives. March 2015.

[30] Christopher Nowakowski, Steven E Shladover, Xiao-Yun Lu, Deborah Thompson, and Aravind Kailas. Cooperative adaptive cruise control (CACC) for truck platooning: Operational concept alternatives. March 2015.

[31] Stefania Santini, Alessandro Salvi, Antonio Saverio Valente, Antonio Pescapè, Michele Segata, and R Lo Cigno. A consensus-based approach for platooning with inter-vehicular communications. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 1158–1166. IEEE, May 2015.

[32] Alan Ali, Gaetan Garcia, and Philippe Martinet. The flatbed platoon towing model for safe and dense platooning on highways. *IEEE Intelligent transportation systems magazine*, 7(1):58–68, January 2015.

[33] Jeroen Ploeg, Bart TM Scheepers, Ellen Van Nunen, Nathan Van de Wouw, and Henk Nijmeijer. Design and experimental evaluation of cooperative adaptive cruise control. In *2011 14th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pages 260–265. IEEE, October 2011.

[34] IEEE Guide for Wireless Access in Vehicular Environments (WAVE) - Architecture. *IEEE Std 1609.0-2013*, pages 1–78, March 2014.

[35] Michael McGurrin et al. Vehicle information exchange needs for mobility applications. Technical report, United States. Joint Program Office for Intelligent Transportation Systems, February 2012.

[36] IEEE. IEEE 1609.2: IEEE Standard for Wireless Access in Vehicular Environments - Security Services for Applications and Management Messages. January 2016.

[37] Tim Leinmüller, Levente Buttyan, Jean-Pierre Hubaux, Frank Kargl, Rainer Kroh, Panos Papadimitratos, Maxim Raya, and Elmar Schoch. SEVECOM - Secure Vehicle Communication. June 2006.

[38] Panagiotis Papadimitratos, Levente Buttyan, Tamás Holczer, Elmar Schoch, Julien Freudiger, Maxim Raya, Zhendong Ma, Frank Kargl, Antonio Kung, and Jean-Pierre Hubaux. Secure vehicular communication systems: design and architecture. *IEEE Communications Magazine*, 46(11):100–109, November 2008.

[39] Frank Kargl, Panagiotis Papadimitratos, Levente Buttyan, Michael Müter, Elmar Schoch, Bjorn Wiedersheim, Ta-Vinh Thong, Giorgio Calandriello, Albert Held, Antonio Kung, et al. Secure vehicular communication systems: implementation, performance, and research challenges. *IEEE Communications magazine*, 46(11):110–118, November 2008.

[40] Maxim Raya, Panos Papadimitratos, and Jean-Pierre Hubaux. Securing vehicular communications. *IEEE wireless communications*, 13(5):8–15, November 2006.

[41] ETSI TS 102 940 V1.3.1. Intelligent Transport Systems (ITS); Security; ITS communications security architecture and security management. 2018.

[42] Mohammad Khodaei, Hongyu Jin, and Panos Papadimitratos. Towards Deploying a Scalable & Robust Vehicular Identity and Credential Management Infrastructure. In *Proceedings of the IEEE Vehicular Networking Conference 2014 (VNC 2014)*, pages 33–40, Paderborn, Germany, December 2014.

[43] M. Khodaei and P. Papadimitratos. Evaluating On-demand Pseudonym Acquisition Policies in Vehicular Communication Systems. In *Proceedings of the First International Workshop on Internet of Vehicles and Vehicles of Internet (IoV/VoI)*, Paderborn, Germany, July 2016.

[44] ETSI TR 103 415 V1.1.1. Intelligent Transport Systems (ITS); Security; Pre-standardization study on pseudonym change management. April 2018.

[45] Mohammad Khodaei and Panos Papadimitratos. Efficient, Scalable, and Resilient Vehicle-Centric Certificate Revocation List Distribution in VANETs. In *Proceedings of the 2018 ACM conference on Security and privacy in wireless & mobile networks*, pages 172–183, Stockholm, Sweden, June 2018.

[46] Mohammad Khodaei and Panos Papadimitratos. Scalable & Resilient Vehicle-Centric Certificate Revocation List Distribution in Vehicular Communication Systems. *IEEE Transactions on Mobile Computing (IEEE TMC)*, 20(7):2473–2489, July 2021.

[47] Ahmed Abdo, Sakib Md Bin Malek, Zhiyun Qian, Qi Zhu, Matthew Barth, and Nael Abu-Ghazaleh. Application level attacks on Connected Vehicle Protocols. pages 459–471, September 2019.

[48] Seyhan Ucar, Sinem Coleri Ergen, and Oznur Ozkasap. IEEE 802.11 p and visible light hybrid communication based secure autonomous platoon. *IEEE Transactions on Vehicular Technology*, 67(9):8667–8681, May 2018.

[49] Rens van der Heijden, Stefan Dietzel, and Frank Kargl. Misbehavior detection in vehicular ad-hoc networks. *1st GI/ITG KuVS Fachgespräch Inter-Vehicle Communication. University of Innsbruck*, pages 23–25, February 2013.

[50] Steven So, Prinkle Sharma, and Jonathan Petit. Integrating plausibility checks and machine learning for misbehavior detection in VANET. In *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 564–571. IEEE, December 2018.

[51] Matthias Matousek, Mahmoud Yassin, Rens van der Heijden, Frank Kargl, et al. Robust detection of anomalous driving behavior. In *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*, pages 1–5, Porto, Portugal, June 2018. IEEE.

[52] Joseph Kamel, Mohammad Raashid Ansari, Jonathan Petit, Arnaud Kaiser, Ines Ben Jemaa, and Pascal Urien. Simulation framework for misbehavior detection in vehicular networks. *IEEE transactions on vehicular technology*, 69(6):6631–6643, April 2020.

[53] R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–45, March 1960.

[54] Simon J. Julier and Jeffrey K. Uhlmann. New extension of the Kalman filter to nonlinear systems. 3068:182 – 193, July 1997.

[55] ETSI TS 122 186 V16.2.0. 5G; Service requirements for enhanced V2X scenarios (3GPP TS 22.186 version 16.2.0 Release 16). November 2020.

[56] Rajesh Rajamani, Han-Shue Tan, Boon Kait Law, and Wei-Bin Zhang. Demonstration of integrated longitudinal and lateral control for the operation of automated vehicles in platoons. *IEEE Transactions on Control Systems Technology*, 8(4):695–708, July 2000.

[57] Lawrence R Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.

[58] G David Forney. The viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, March 1973.

[59] Leonard E Baum, Ted Petrie, George Soules, and Norman Weiss. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. *The annals of mathematical statistics*, 41(1):164–171, August 1970.

[60] Gideon Schwarz et al. Estimating the dimension of a model. *Annals of statistics*, 6(2):461–464, March 1978.

[61] Michele Segata, Stefan Joerer, Bastian Bloessl, Christoph Sommer, Falko Dressler, and Renate Lo Cigno. Plexe: A platooning extension for veins. pages 53–60, August 2014.

[62] Open Source Vehicular Network Simulation Framework. `veins.car2x.org`, July 2019.

[63] Michael Behrisch, Laura Bieker, Jakob Erdmann, and Daniel Krajzewicz. SUMO - Simulation of Urban MObility: An Overview. In *Intern. Conference on Advances in System Simulation*, Barcelona, Spain, October 2011.

TRITA-EECS-EX-2021:785