Detection of False Data Injection Attack in Connected Vehicles via Cloud-based Sandboxing

Final Report

by

Pierluigi Pisu, Ph.D., Clemson University
Gurcan Comert, Ph.D., Benedict College
Chunheng Zhao, Ph.D. student, Clemson University

Contact information
Pierluigi Pisu, Ph.D.
4 Research Drive, Greenville, SC 29607
Clemson University
Phone: (864) 283-7227; *E-mail:* pisup@clemson.edu

October 2020

Center for Connected Multimodal Mobility (C²M²)

200 Lowry Hall, Clemson University
Clemson, SC 29634

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The report is funded, partially or entirely, by the Center for Connected Multimodal Mobility (C^2M^2) (Tier 1 University Transportation Center) Grant, which is headquartered at Clemson University, Clemson, South Carolina, USA, from the U.S. Department of Transportation's University Transportation Centers Program. However, the U.S. Government assumes no liability for the contents or use thereof.

Non-exclusive rights are retained by the U.S. DOT.

ACKNOWLEDGMENT

The authors would like to acknowledge the Center for Connected Multimodal Mobility (C^2M^2), which is a Tier 1 University Transportation Center, for supporting this research. The authors also would like to acknowledge IBM for their support for Cloud computing resources.

Technical Report Documentation Page

1. Report No.	**2. Government Accession No.**	**3. Recipient's Catalog No.**

4. Title and Subtitle Detection of False Data Injection Attack in Connected Vehicles via Cloud-based Sandboxing	**5. Report Date** Oct. 2020 **6. Performing Organization Code**
7. Author(s) Pierluigi Pisu, Ph.D.; ORCID: 0000-0003-4266-1336 Gurcan Comert, Ph.D.; ORCID: 0000-0002-2373-5013 Chunheng Zhao, student; ORCID: 0000-0002-3121-4779	**8. Performing Organization Report No.**
9. Performing Organization Name and Address Department of Automotive Engineering Clemson University 4 Research Drive, Greenville, SC 29681	**10. Work Unit No.** **11. Contract or Grant No.** 69A3551747117
12. Sponsoring Agency Name and Address Center for Connected Multimodal Mobility (C^{2}M^2) Clemson University 200 Lowry Hall, Clemson, SC 29634	**13. Type of Report and Period Covered** Final Report (December 2018 –July 2020) **14. Sponsoring Agency Code**

15. Supplementary Notes

16. Abstract

In recent years, developments in vehicle-to-everything communication (V2X) have steadily increased in applications such as platooning, collision avoidance, and routing algorithm. V2X provides vehicles with long range information regarding traffic congestion and routing, but also short and mid-range information allowing cooperative adaptive cruise control, automatic collision warnings, and others. Despite being potentially beneficial in several aspects, such interdependency poses a set of specific challenges from a safety and reliability standpoint due to the possibility of cyber-attacks aimed at influencing the behavior of the vehicles. In this project, a Cloud-based method to detect the false data injection attack on Connected Autonomous Vehicles (CAVs) is presented. The sandboxing concept utilized in this report comes from computer security and it is recast in a control framework as a way to isolate and evaluate the data exchanged by the CAVs affecting the vehicle control system. Numerical experiments are conducted to show the effectiveness of the approach using microscopic traffic simulation. Our results are summarized as follows: (i) both two proposed data fusion algorithms with different architecture are able to improve the localization results of connected and autonomous vehicles; (ii) both two proposed attack detection schemes are able to detect false data injection attacks in platooning scenario and rerouting scenario, respectively.

17. Keywords Connected and Autonomous Vehicles; Security; False Data Injection; Particle Filters; Attack Detection.	**18. Distribution Statement** No restrictions

19. Security Classif. (of this report)	**20. Security Classif. (of this page)**	**21. No. of Pages**	**22. Price**
Unclassified	Unclassified	33	NA

Table of Contents

LIST OF TABLES

LIST OF FIGURES

EXECUTIVE SUMMARY

Developments in vehicle-to-vehicle (V2V) and vehicle-to-infrastructure communications (V2I) have been steadily increasing. Considerable research has been conducted for fully autonomous vehicles on platooning, collision avoidance, intersection control, and similar intelligent transportation applications. According to these studies, V2V and V2I communications will be extensively implemented in future mobility, providing the vehicles with long range information regarding the presence of road incidents, traffic congestion, surface conditions, routing, but also short and mid-range information allowing cooperative adaptive cruise control, automatic collision warnings, cooperation at intersections, and others. Despite being potentially beneficial in several aspects (e.g., traffic management, reduction of fuel consumption, and driver assistance) such interdependency poses a set of specific challenges from a safety and reliability standpoint, due to the possibility of cyber-attacks aimed at influencing the behavior of the vehicles by exploiting their connectivity. Improving communication security can help to prevent this problem, however, it has been shown that addressing cybersecurity issues exclusively from the cyber side of the CPS presents several drawbacks. In the literature, a lot of solutions have been developed considering different attacks on multiple connected vehicle applications. However, the scope of this research is developing a more general and scalable technique that will be applicable to a vast set of cooperation-based control algorithms. We envision that our technique will play an important role in accelerating the spreading of CAVs and cooperation-related driver assistance services.

The overall vision of this project is in the development of an attack detection algorithm for cooperative CAV resilient to false data injection attacks and therefore capable of satisfying more stringent system safety and performance requirements. More specifically, we borrow the sandboxing concept from computer security and recast it in the control framework as a way to isolate and evaluate the data exchanged by the CAVs affecting the vehicle control system. The main objective of this project is, therefore, to address the challenge by exploiting cloud computing techniques and sandboxing approach to achieve higher resilience to a particular type of cyber-attack known as false data injection. In order to achieve such a goal, it is required to efficiently manage the data flow to the cloud and use optimized computational algorithms to enable fast calculations so that the critical sandboxing step can be performed accounting for the real-time constraint imposed by the physical system. The main activities for this project are summarized as follows.

- Create a data fusion algorithm that can yield an accurate estimation of current traffic conditions. Data fusion here involves feasible information given historical data, different sensors, and traffic parameter states observed.
- Develop the decision logic algorithm that will compare predicted outcomes and will allow distinguishing between trustworthy and malicious information. This requires defining appropriate evaluation metrics and designing thresholds.
- Build a platooning scenario and a malicious vehicle rerouting scenario. Validate the proposed approaches using the built scenario.

In total, this project aims at developing a cloud-based sandboxing technique that will allow CAVs to safely operate even in corrupted conditions when malicious data is injected into the communication network. Given a foreseeable future in which CAV technology is expected to enter the market, the proposed research addresses the problem of improving the resilience of CAVs to the possibility of cyber-attacks aimed at impairing or anyway affecting their behavior by injecting false data in the shared information flow.

The main results of this project include:

- Two different data fusion algorithms with different architecture are presented. The methodology leverages vehicles' connectivity and Particle Filters for vehicle state estimation and attack detection. The proposed approaches combine Particle Filters and vehicle-to-vehicle communication in order to fuse the location and speed information of multiple vehicles. Both of the two data fusion algorithms are able to provide a better estimation of the vehicle’s state than the vehicle’s onboard sensor measurements.
- The results of the data fusion algorithm are used to construct the decision-making scheme in order to identify and isolate an attacker. Two decision schemes are developed based on the two data fusion algorithms. They leverage the knowledge of diagnostics and consensus decision making. Both of the two detection schemes are able to detect and isolate false data injection attacks.
- A platooning scenario and a vehicle navigation routing scenario are built and used to validate the proposed algorithms in the Cloud server. The Cloud supervises the vehicle's operations by collecting, fusing, and processing their information, and by performing a sandbox simulation which allows to filter the information and so to feed each vehicle only with the part that can be safely utilized.

CHAPTER 1

Introduction

Developments in vehicle-to-vehicle (V2V) and vehicle-to-infrastructure communications (V2I) have been steadily increasing. The automotive vehicle to everything (V2X) market is projected to grow from USD 689 million in 2020 to USD 12,589 million by 2028, at a compound annual growth rate (CAGR) of 44.2% (Markets and Markets (2020)). Considerable research has been conducted for fully autonomous vehicles on platooning, collision avoidance, intersection control, and similar intelligent transportation applications (Lu et al. (2014); Dey et al. (2015); Kong et al. (2017)). According to these studies, V2V and V2I communications will be extensively implemented in future mobility, providing the vehicles with long range information regarding the presence of road incidents, traffic congestion, surface conditions, routing, but also short and mid-range information allowing cooperative adaptive cruise control, automatic collision warnings, cooperation at intersections, and others. The future transportation network can be modeled as a cyber-physical system (CPS) in which communication networks and transportation infrastructures are strictly interconnected (Fallah et al. (2010); Work et al. (2008)). Despite being potentially beneficial in several aspects (e.g., traffic management, reduction of fuel consumption, and driver assistance) such interdependence poses a set of specific challenges from a safety and reliability standpoint, due to the possibility of cyber-attacks. Such attacks would aim at influencing the behavior of the vehicles by exploiting their connectivity. This means that each connected and autonomous vehicle (CAV) would face the issue of deciding whether to trust or not the information that it is receiving from the CAV network. False data injection, packet dropping, and forced network congestion are just some of the possible techniques that can be exploited to manipulate the behavior of control systems based on vehicle connectivity, such as Cooperative Adaptive Cruise Control, Advanced Driver-Assistance Systems, and intelligent signal control (Petit et al. (2014); Chowdhury et al. (2019)).

In this project, the main goals are summarized as

- Create a data fusion algorithm that can yield an accurate estimation of current traffic conditions. Data fusion here is going to involve feasible information given historical data, different sensors, and traffic parameter states observed.
- Develop a decision logic algorithm that will compare fusion outcomes and will allow distinguishing between trustworthy and malicious information. This will require defining appropriate evaluation metrics and designing thresholds.
- Validate the proposed approach and the behavior of the vehicles under false data injection attack in urban scenarios (e.g., vehicle platoons or vehicle routing) and uncertainties in localization in the case of fully connected and different CAV penetration rate scenarios.

The remainder of this report is organized as follows. Chapter 2 provides a literature review of the cyber-attacks in connected and autonomous vehicles. Chapter 3 discussed the proposed attacks detection approach: system model, particle filter-based data fusion algorithm, attack detection scheme, and Cloud-based architecture. Chapter 4 presents numerical experiments on microscopic traffic simulation. Lastly, Chapter 5 provides concluding remarks and future works.

CHAPTER 2

Literature Review

In the literature, a lot of solutions have been developed considering different attacks on multiple connected vehicle applications. Sharma et al. (2017) propose an artificial intelligence (AI) predictive algorithm based on Bayesian theory like Kalman and Particle Filter along with generic filters to detect spoofed messages with robustness to denial of service (DoS) attacks. Singh et al. (2015) explore novel secure cloud networks for connected vehicle services to enhance the transportation system reliability. In their paper, cloud networks will provide access and control of multiple vehicles on the road to improve the safety of the passengers and optimize the service of the transportation system in a real-time traffic application. Moreover, a secure protocol for exchanging inter-vehicular messages which rely on the consensus strength (blockchain) is also presented (Calvo et al. (2018)). This implementation provides the advantages of being decentralized, anonymous, and forgery-proof, i.e., the previously accepted values cannot be modified. Ali Alheeti et al. (2016) propose misuse and anomaly detection or hybrid Intrusion Detection System (IDS) based on backpropagation artificial neural networks (ANNs) to predict attacks on the external communications of self-driving and semi self-driving vehicles. Some research has also been done on securing vehicular ad-hoc networks (VANETs) using IDS (Li et al. (2015); Liang et al. (2019)). However, IDS faces the problem that it may spend too much time on detection. In order to reduce the overhead and detection time in IDS, Sedjelmaci et al. consider a trade-off between intrusion detection rate and overhead using a Bayesian game model (Sedjelmaci et al. (2016)).

These defense approaches try to secure connected vehicles only using cyber knowledge. However, as connected vehicles could be modeled as a cyber-physical system (CPS), it has been shown that addressing cybersecurity issues exclusively from the cyber side of the CPS presents several drawbacks, for example, software patching and frequent updates are not well suited for control systems (Cardenas et al. (2009)). Therefore, controlled-based solutions have to be addressed in order to solve the problem. Biron et al. (2017) showed the possible risks related to different types of cyber-attacks on vehicle platoons via Cooperative Adaptive Cruise Control (CACC) applications (Biron et al. (2017); Biron et al. (2018); Rayamajhi et al. (2018)). CACC aims at improving highway capacity and fuel consumption, however, attacks such as DoS and false data injection induce severe performance degradation in the whole system with increased risk of collision. Apart from these control-based approaches, a deep neural network and Kalman filter combined approach is proposed by Van et al. (2019) to detect and identify anomalous behavior in CAV.

Petrillo et al. (2017) designed a collaborative, consensus-based control strategy that can both counteract communication impairments, such as the usual time-varying communication delays, and mitigate the effects of the message falsification attack on the platoon behavior. Fiengo et al. (2016) enhanced the very recent cooperative cruise control algorithm for autonomous and connected vehicles presented in \cite{zhang2016motif}, by explicitly considering heterogeneous and time-varying delays due to both communication and sensing. Furthermore, some research has been done on the detection of malicious cyber-attacks for cooperative positioning (Mousavinejad et al. (2019); Kong et al. (2017); Heng et al. (2014)). However, these approaches are focusing on one typical connected vehicle application like CACC. For other attacking scenarios, Zeng et al. (2017) proposed an attack model in road navigation scenarios and developed a complete framework to analyze, simulate and evaluate the spoofing attacks under practical constraints. Lin et al. (2018) investigated security issues of route guidance schemes via modeling and analysis of data integrity attacks on the route guidance process and then developed corresponding mitigation mechanisms to combat the investigated attack.

In this study, the main focus is on false data injection attacks, and a methodology via sandboxing technology is proposed to assess the trustworthiness of information exchanged by CAVs. The main objective is, therefore, to address the challenge by exploiting cloud computing techniques and sandboxing approach to achieve higher resilience to a particular type of cyber-attack known as false data injection. In order to achieve such a goal, it is required to efficiently manage the data flow to the cloud and use optimized computational algorithms to enable fast calculations so that the critical sandboxing step can be performed accounting for the real-time constraint imposed by the physical system.

CHAPTER 3

Research Approach

3.1 Overall Flow of Work

The research methods and overall flow of work activities are summarized as follows:

- First, two different data fusion algorithms with different architectures are developed in order to fuse data between connected vehicles. The data includes vehicles' position and velocity.
- Second, two attack detection schemes are developed, which corresponds to the two data fusion algorithms, to detect false data injection attacks.
- Third, a platooning scenario and a rerouting scenario are developed in VISSIM.
- Fourth, the proposed attack detection approaches are validated in the built scenarios.

3.2 Cloud-based Sandboxing

In Information Technology (IT), the term sandboxing indicates the evaluation of the effects of untested and untrusted code in a testing environment before making it available to the actual system, therefore protecting critical resources from potential damages. In control theory, the same principle is used to create a framework that allows handling unknown or untrusted controllers, measures, or information in general. The main idea consists of integrating the transmitting information in a controlled environment (in this case the Cloud). The outcome is then used to evaluate whether or not to trust information. The main purpose of the Cloud consists of gathering information from all the partakers at the road-level and generating a flow of trustworthy information for every CAV which ensures the safety of the vehicles (Bak et al. (2011)).

In recent years, there's been a lot of developments in cloud computing technologies. Compared with traditional computing solutions, cloud computing has the following main advantages (Cole, (2019); Salesforce, (2018)).

- Reduced cost: the reduction of numbers of servers, the software cost can significantly reduce IT costs without affecting an organization's IT capabilities.
- Improved mobility: data is available to employees no matter where they are in the world and when they want to access it.
- Flexible capacity: the cloud is a flexible facility that can be turned up, down, or off depending on specific applications. Capacity can increase as the need for computing increases and shrink when the task is over.
- Enhanced security: a cloud host's full-time job is to carefully monitor data security. The encryption of data being transmitted to cloud servers and stored in databases makes cloud computing a good solution to keep sensitive information offsite.

A Virtual Machine (VM) is a virtual representation, or emulation, of a physical computer. VM is the basis and fundamental unit of cloud computing, which enables dozens of different types of applications and workloads to run (Palmer, (2018)). Existing major cloud providers like Amazon Web Services (AWS) and Microsoft Azure provide multiple choices of VM according to user's application and workload. Benefiting from the powerful VMs provided by the cloud platform, the attack detection task could be handled on the cloud.

Following the Simplex architecture in (Sha et al. (2001)), our proposed structure requires the definition of a Cooperative Controller (CC) and an Attack Detection Unit (ADU) as shown in Figure 1. The Cooperative Controller is a supervisory controller with enhanced performance compared to a standard controller. The CC relies on shared CAVs information to make decisions. In a platooning scenario, a Cooperative Adaptive Cruise Control (CACC) could be a Cooperative Controller which enhances performances by improving the string stability. An Adaptive Cruise Control (ACC), on the other hand, is a standard controller which cannot exhibit string stability. In a rerouting scenario, a connected vehicle-based dynamic routing algorithm could be a Cooperative Controller (CC) which includes real-time traffic data to route selection to reduce travel cost. An offline static routing algorithm is a standard controller which cannot utilize real-time traffic data, thus cannot exhibit best routes selection based on changing traffic condition. The control action generated by the CC leads to better performance. However, when specific security conditions are not met, e.g., in the case of false data injection, the ADU should detect it and prevent such data from being used by the Cooperative Controller. Once the system's security is restored, the ADU allows the information to be utilized again in order to achieve the best performance.

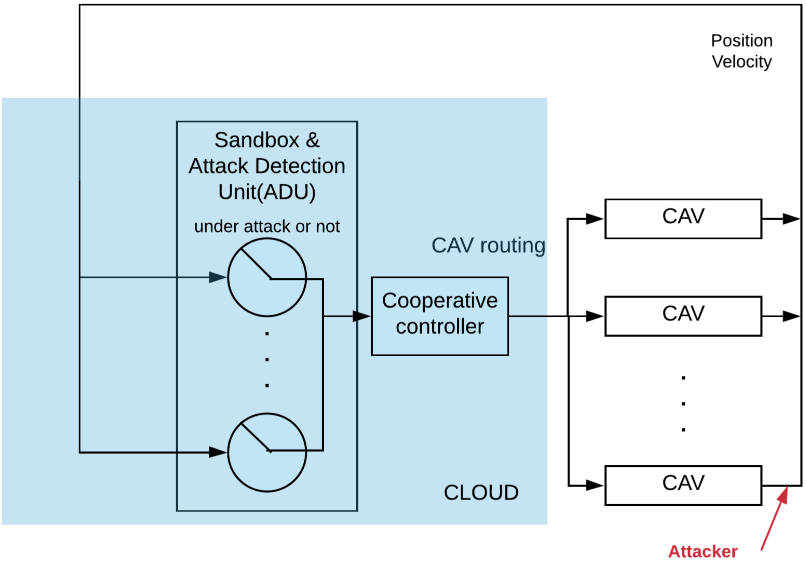


Figure 1: Architecture of proposed cloud-based sandboxing method

In particular, this approach will require the Attack Detection Unit to compare the future state of the single vehicle with the estimated evolution from the CAVs network. Such estimation will be performed within a cloud-based data collection and sensor-fusion environment. Benefiting from the powerful computational capability of the Cloud and its advantage in security, the ADU should be able to handle the task in real-time with a high level of reliability.

In case of the communication delay and channel modeling, for the platooning scenario, it is assumed that as long as the communication delay or latency between Cloud and vehicles is acceptable, a Cloud-based platooning scenario would be reasonable. An acceptable delay has to be a fraction of time-to-collision. Recent work of Deng et al. (2020) reports that 0.4 s latency would be acceptable in a Cloud-based speed advisory application in an arterial setting (Perimeter Rd, Clemson, SC) tested with actual vehicles. For instance, vehicles traveling at 53 km/h with 6 meters (m) minimum following distance would require less than 0.4 s latency. On freeways at higher speeds, longer following distance or lower latency would be needed. Although, a simplistic reasoning can be given, the modeling of the Cloud communication channel is out of the scope of this report, but several works have been proposed to embed V2V in Cloud computing (Dey et al. 2016, Hussain et al. 2015, Nasimi et al. 2020, Mon et al. 2018, Chang et al. 2017}. As one of the goals of this report is to develop a general defense for various CAV applications, the platooning scenario in this paper is set to regular CAV platoons. In terms of the rerouting scenario, there is an increasing demand for streaming more data than a platooning scenario, the modeling of Cloud communication channel is also out of the scope of this project but the feasibility of streaming huge data through the Cloud with low latency has been validated in the literature, for example, 5G with vehicular Cloud (Balasubramanian et al. 2020).

More specifically, CAVs can share their current position and speed with the Cloud via vehicle to infrastructure (V2I) network. The Cloud supervises the CAVs operations by collecting, fusing, processing their information, and by performing a sandbox simulation which allows to filter the information and so to feed each CAV only with the part that can be safely utilized. The connected vehicles communicate with the Cloud server and publish information with a unique vehicle ID. The unconnected vehicle's information is assumed to be generated from on-board sensors of connected vehicles using a multi-source data association method so that each unconnected vehicle is assigned with a specific vehicle ID. The information that CAVs upload to the Cloud server includes their position, velocity, acceleration, relative distance, and relative speed from neighboring vehicles. It is assumed that only vehicles with a radius R from a CAV can be sensed. Although false data injection, package dropping, forced network congestion, and some other possible techniques can be exploited to maliciously affect the behavior of platooning or rerouting scenarios, only, the false data injection attack is considered here. The attack is assumed to be on the CAVs in the procedure of publishing information with Cloud server as shown in Figure 1.

3.3 System Model

In order to obtain accurate localization information which will be utilized in the CC, the idea of cooperative localization is introduced here. The proposed data fusion scheme incorporates a Particle Filter with Cloud communication. The approach integrates the multi-source data and cooperatively improves the accuracy of the localization information of connected vehicles on the road. There is one Particle Filter running for each vehicle that aims to fuse its onboard sensor information with the received information.

To represent the vehicle motion model, a simple steering and driving model that uses gyroscopes and accelerometers to find the vehicles' yaw rate and acceleration is considered. Therefore, the current input of the system can be defined by a pose vector $u_t = [\dot{\psi} \quad a]$, where $\dot{\psi}$ and a are current yaw rate and acceleration, respectively. The state transition equation of the vehicle system is shown in Eq. (1).

$$X_t = f(u_t, X_{t-1}) = \begin{cases} x_t = x_{t-1} + v_t \cdot cos\psi \cdot \Delta t + \epsilon_{t1} \\ y_t = y_{t-1} + v_t \cdot sin\psi \cdot \Delta t + \epsilon_{t2} \\ v_t = v_{t-1} + a_t \cdot \Delta t + \epsilon_{t3} \\ \psi_t = \psi_{t-1} + \dot{\psi}_t \cdot \Delta t + \epsilon_{t4} \end{cases} \tag{1}$$

where, $X_t = [x_t \quad y_t \quad v_t \quad \psi_t]^T$ is the state of the vehicle at time t, Δt is time step, and ϵ_{ti} (i = 1 ,... 4) is a set of random samples drawn from $N(0, \sigma_a^2)$ representing system noise.

Regarding cooperative localization, apart from obtaining the localization information directly from on-board GPS, the ego vehicles also estimate their location and speed based on their neighbors' GPS data, the relative distance, and speed between ego vehicle and its neighbors. The relative distance and speed could be obtained through lidar and radar measurements. A vehicle and its neighbors at time t are represented by i_t and $N_t^{(i)}$, respectively. Assuming that $j \in N_t^{(i)}$, its estimation about the location and velocity of i is expressed by

$$\begin{cases} x_t^{(ji)} = x_t^{(j)} + d_t^{(ji)} cos(\gamma_t^{(ji)}) \\ y_t^{(ji)} = y_t^{(j)} + d_t^{(ji)} sin(\gamma_t^{(ji)}) \\ v_t^{(ji)} = v_t^{(j)} + s_t^{(ji)} sin(\gamma_t^{(ji)}) \end{cases} \tag{2}$$

where, $x_t^{(ji)}$, $y_t^{(ji)}$, and $v_t^{(ji)}$ are the estimation of i's location and velocity in the coordinate frame of j. $d_t^{(ji)}$, $s_t^{(ji)}$, and $\gamma_t^{(ji)}$ are relative distance, relative velocity, and the angle between two vehicles at time t using lidar and radar, respectively. $x_t^{(j)}$, $y_t^{(j)}$, and $v_t^{(j)}$ are the estimation of i's neighbor j's location and velocity. As it is difficult to infer the neighboring vehicle's yaw angle from the ego vehicle's onboard sensor, to simplify the problem, a false yaw angle attack is not considered.

For the data fusion scheme, full state observation is assumed which contains the measurement of the current position, speed, and yaw angle. Therefore, the GPS and onboard speed sensor readings are:

$$Z_t = [\tilde{x}_t \quad \tilde{y}_t \quad \tilde{v}_t \quad \tilde{\psi}_t]^T \tag{3}$$

The observation equations for this model are

$$Z_t = g(X_t, n_t) = \begin{cases} \tilde{x}_t = x_t + n_t^{(1)} \\ \tilde{y}_t = y_t + n_t^{(2)} \\ \tilde{v}_t = v_t + n_t^{(3)} \\ \tilde{\psi}_t = \psi_t + n_t^{(4)} \end{cases} \tag{4}$$

where, n_t is a set of random samples drawn from $N(0, \sigma_n^2)$ representing measurement noise. Note that for different states, measurement noises vary.

The routing algorithm for connected vehicles is adapted and simplified from (Tian et al. (2015)). The criterion for the best route is the general travel cost. A route is a sequence of edges that describes a path through the network. For each of the edges in the network, the general cost of that edge i for period k is computed as a weighted sum of travel time T_i^k and travel distance d_i^k.

$$C_i^k = \alpha \cdot T_i^k + \beta \cdot d_i^k \tag{5}$$

where the travel distance is determined by the geometry of the edges and travel time is computed depending on the traffic situation. The coefficients α and β can be defined by the user. During a simulation, travel times are measured for each edge in the network. All vehicles that leave the edge report the time they have spent on the edge. All travel times during one evaluation interval

k are averaged and thus form the measured travel time for that edge. The general cost C_j^k for a route, j is simply defined as the sum of the general costs C_i^k of all its edges i:

$$C_j^k = \sum_{i \in j} C_i^k \tag{6}$$

Then the route with minimum cost will be selected.

3.4 Particle Filter based Data Fusion Algorithm I (two-stages)

In this section, this first data fusion algorithm is designed for the platooning scenario. In our proposed method, Monte Carlo Localization (MCL) algorithm, which is based on Particle Filter (PF), is utilized. The core of MCL is to represent the belief by a set of weighted samples (particles), where each one of those particles represents a full state, here including the vehicle's location, velocity, and yaw angle. Particles at places other than the most likely states gradually disappear, thus, particles at better places keep converging to form a Gaussian distribution of their weights.

The following data fusion algorithm is a modification of the PF in (Golestan et al. (2015)). Here, the neighboring vehicles around one ego vehicle could also be regarded as samples (particles) because each of them could also provide an estimate of the ego vehicle’s states based on Eq. (2). Thus, the idea is to let PF update the weights for each neighboring vehicle according to the likelihood. Neighboring vehicles that give a bad estimation of the ego vehicle state would be assigned lower weights. Instead of just using neighboring vehicles as particles to calculate the ego vehicle's state, which may lead to the unstable condition of particle filter due to the insufficient and changing number of particles, the proposed algorithm integrates two stages of filters and uses a fixed number of particles to calculate the final state. In this method, data fusion is implemented by involving neighboring vehicles' measurements as particle states in the particle filter (i.e., regarding neighboring vehicles as particles). More details are shown in Figure 2 and will be discussed in the following paragraphs.

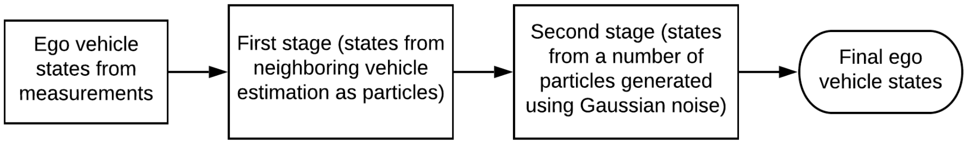

Ego vehicle states from measurements
First stage (states from neighboring vehicle estimation as particles)
Second stage (states from a number of particles generated using Gaussian noise)
Final ego vehicle states

Figure 2: Two-stages architecture of the proposed data fusion algorithm

A particle filter coupled with stratified sampling is utilized with the benefits of reducing the variance and fusing the data in a probabilistic way. Using a stratified sampling technique, based on the updated weight of each source (neighboring vehicles), allows assigning particles proportional to the calculated weight so that those particles can update their weights based on the belief of that source (each neighboring vehicle). If one neighboring vehicle has less weight, a lower number of particles will be assigned to it; therefore, its belief does not affect the prediction too much. Using this technique, the aim is to have particles’ weight based on all neighboring vehicles and proportional to their reliability. The state transition equation for the cooperative vehicle localization system can be represented by Eq. (7).

$$X_t^{(ji)} = \begin{cases} x_{t-1}^{(ji)} + v_t \cdot cos\psi \cdot \Delta t \\ y_{t-1}^{(ji)} + v_t \cdot sin\psi \cdot \Delta t \\ v_{t-1}^{(ji)} + a_t \cdot \Delta t \\ \psi_{t-1} + \dot{\psi}_t \cdot \Delta t \end{cases} \tag{7}$$

As shown in Figure 2, there are two processing stages: in the first stage, the weight for each neighboring vehicle, which contributes to defining the final subject vehicle i's location, is calculated. These weights are calculated using Eq. (8) – (10) in which, for each neighboring vehicle j, the weight is proportional to its previous weight and the likelihood of vehicle i regarding the location of vehicle j (current observation likelihood $p(Z_t|X_t^{(ji)})$. Z_t is the observation model in Eq. (4).

$$\omega_{t-1}^{(ji)} = \frac{\tilde{\omega}_{t-1}^{(ji)}}{\sum_{j=1}^{N} \tilde{\omega}_{t-1}^{(ji)}} \tag{8}$$

$$\tilde{\omega}_t^{(ji)} = \omega_{t-1}^{(ji)} \cdot p(Z_t|X_t^{(ji)}) \tag{9}$$

$$p(Z_t|X_t^{(ji)}) = \frac{1}{\sqrt{(2\pi)^4 \prod_{n=1}^{4} \sigma_n(n)}} \cdot exp(-\frac{1}{2} \cdot (e_t^j)^2) \tag{10}$$

where, $N_t^{(j)}$ denotes neighboring vehicles set at time t and ω^{ji} is the weight assigned to the belief of vehicle i in accordance to the location of vehicle j; e^{ji} is the normalized error between the actual measurement of i and estimation of i's location in accordance to j's location at time t, which is calculated using Eq. (11).

$$e_t^j = \sqrt{\sum_{n=1}^{4} \left(\frac{Z_t(n) - X_t^{(ji)}(n)}{\sigma_n(n)}\right)^2} \tag{11}$$

where n represents the index of every single state in the observation equation or state transition equation. Here four states are considered, therefore $= \{1,2,3,4\}$.

In the second stage, a total of M particles are assigned to each neighboring vehicle using stratified sampling based on weights. These particles can update their weights based on the belief of corresponding neighboring vehicles using the state transition model in Eq. (1). Lastly, all weights and states are calculated for the central target vehicle. The expected value is approximated by the weighted sum as

$$E[X_t] \simeq \sum_{m=1}^{M} X_t^{(m)} \cdot \omega_t^{(m)} \tag{12}$$

3.5 Particle Filter based Data Fusion Algorithm II (multi-sensor)

In this section, this second data fusion algorithm is designed for the rerouting scenario. In this proposed method, the core architecture is still based on a particle filter. Different from the first two-stages architecture where neighboring vehicles are regarded as particles, this second architecture uses neighboring vehicles as additional measurements in the observation equation, which could be regarded as a multi-sensor architecture. In this method, data fusion is implemented by involving neighboring vehicle measurements as additional measurements in the particle filter (i.e., regarding neighboring vehicles as complement sensors).

More specifically, each neighboring vehicle is able to provide an estimate of the ego vehicle's states using Eq. (13) which is based on Eq. (2).

$$z_t^{(ji)} = \begin{cases} x_t^{(ji)} = x_t^{(j)} + d_t^{(ji)} cos(\gamma_t^{(ji)}) + a_{t1} \\ y_t^{(ji)} = y_t^{(j)} + d_t^{(ji)} sin(\gamma_t^{(ji)}) + a_{t2} \\ v_t^{(ji)} = v_t^{(j)} + s_t^{(ji)} sin(\gamma_t^{(ji)}) + a_{t3} \end{cases} \tag{13}$$

where a_{ti} (i = 1 ,... 3) is a set of random samples drawn from $N(0, \sigma_b^2)$ representing measurement noise. Therefore, each neighboring vehicle could be regarded as an additional "sensor" besides the ego vehicle's onboard GPS. And the observation model is not based on measurement from one sensor but from multiple sensors. For time t, a set of measurements Z_t is provided by $j + 1$ sensors:

$$Z_t = \left\{z_t^i\right\} \cup \left\{z_t^{(1i)}, \ldots, z_t^{(ji)}\right\} \tag{14}$$

where z_t^i represents the measurement from the ego vehicle's sensor which is computed based on Eq. (4), $z_t^{(ji)}$ represents the measurements from neighboring vehicles' sensors which are based on Eq. (13) and j denotes the number of neighboring vehicles. Therefore, in total, there are $j + 1$ sets of measurements. As the measurement sets of different sensors are independent, then the observation likelihood $p(Z_t|X_t)$ is computed as

$$p(Z_t|X_t) = p(\left\{z_t^i\right\} \cup \left\{z_t^{(1i)}, \ldots, z_t^{(ji)}\right\} |X_t) = p(z_t^i|X_t) \prod_{j=1}^{N} p(z_t^{(ji)}|X_t) \tag{15}$$

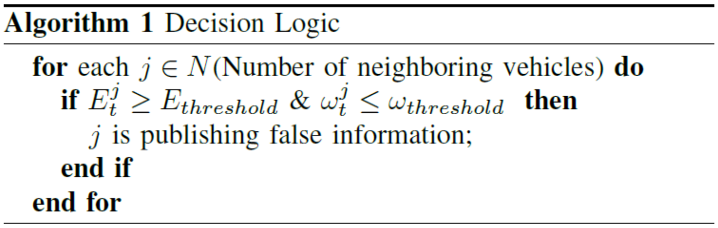
Then the weights for each particle m can be updated using Eq. (16) - (17), and the final output can still be computed using Eq. (12).

$$\omega_{t-1}^{(m)} = \frac{\tilde{\omega}_{t-1}^{(m)}}{\sum_{m=1}^{M} \tilde{\omega}_{t-1}^{(m)}} \tag{16}$$

$$\tilde{\omega}_t^{(m)} = \omega_{t-1}^{(m)} \cdot p(Z_t|X_t^{(m)}) \tag{17}$$

3.6 Attack Detection Scheme for Platooning

The results of the two-stages based data fusion algorithm are going to be compared with the information sent from neighboring vehicles in a platoon to decide whether there is a false data injection attack in the platooning scenario. The logic is shown in Figure 3.



```
Algorithm 1 Decision Logic
for each $j \in N$(Number of neighboring vehicles) do
    if $E_t^j \geq E_{threshold}$ & $\omega_t^j \leq \omega_{threshold}$ then
        $j$ is publishing false information;
    end if
end for
```

Figure 3: Decision logic for attack detection in platooning scenario

In this logic, E_t^j shows the error between estimated results after data fusion and information sent from neighboring vehicle j at time t. ω_t^j represents the weight of that information source (neighboring vehicle j) at time t in Particle Filter. ω_t^j is scalar, but E_t^j is a vector which is defined in Eq. (18). As false yaw angle attack is not considered, there are only three states for E_t^j and $E_{threshold}$. E_t^j exceeds $E_{threshold}$ if at least one element in the vector E_t^j exceeds the corresponding value in the matrix $E_{threshold}$.

$$E_t^j = \begin{bmatrix} e_x \\ e_y \\ e_v \end{bmatrix} \tag{18}$$

where, e_x, e_y, and e_v are the errors for x position, y position, and velocity, respectively.

A false data injection attack on that vehicle is detected when the error is larger than a threshold ($E_{threshold}$) and the weight for that source is lower than a threshold ($\omega_{threshold}$). All the thresholds are selected using the optimal threshold selection method by minimizing the total probability of errors which includes a probability of false alarm (type I) and the probability of a misdetection (type II).

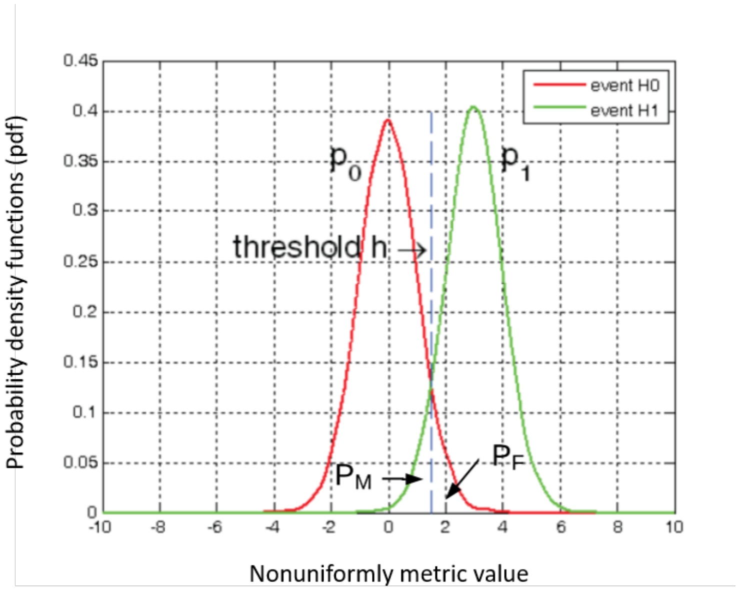


Figure 4: Probability density function (pdf) of the random variable under hypothesis H_0 and H_1

The optimal threshold selection method is based on statistical hypothesis testing as shown in Figure 4. As a decision between two hypotheses (e.g., under attack or no attack) is analyzed, binary hypothesis testing can be applied here. In the binary hypothesis test, residuals corresponding to no attack are assumed to be randomly distributed under the hypothesis H_0, with probability density function (pdf) p_0, while residuals corresponding to an under attack condition are assumed to be randomly distributed under the hypothesis H_1, with probability density function (pdf) p_1. In Eq. (19), P_F refers to the probability that hypothesis H_1 is chosen when H_0 is true (i.e., probability of a false alarm).

$$P_F = \int_h^{+\infty} p_0(x)dx \tag{19}$$

where h is the selected threshold.

In Eq. (20), P_M represents the probability that hypothesis H_0 is chosen when H_1 is true (i.e., probability of a misdetection).

$$P_M = \int_{-\infty}^{h} p_1(x)dx \tag{20}$$

where h is the selected threshold.

A statistical optimal threshold can be obtained by minimizing the total probability of error, $P_E = P_F + P_M$:

$$\min_{h>0}(P_F + P_M) \tag{21}$$

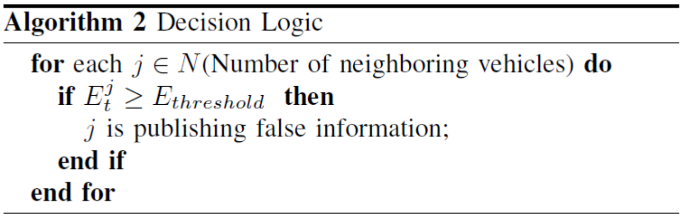
As one neighboring vehicle could be sensed by multiple vehicles and there is a filter for each of these vehicles, there will be multiple results regarding one neighbor vehicle. A decision logic scheme is proposed here that allows analysis of the results and identifies the existence of false data injection attack as well as the source of the attack (Vehicle ID). It is based on an error signature table constructed from the belief information determined from each vehicle with respect to the others including themselves. The beliefs are compared against thresholds and assigned as 0 or 1. An example of a decision scheme for the three vehicles scenario is given in Table 1.

Table 1: Decision scheme for three vehicles scenario

	Attack on 1	Attack on 2	Attack on 3
Decision of 2 from filter on 1	1	1	0
Decision of 3 from filter on 1	1	0	1
Decision of 1 from filter on 2	1	1	0
Decision of 3 from filter on 2	0	1	1
Decision of 1 from filter on 3	1	0	1
Decision of 2 from filter on 3	0	1	1

3.7 Attack Detection Scheme for Rerouting

The results of the multi-sensor-based data fusion algorithm are going to be compared with the information sent from neighboring vehicles to decide whether there is a false data injection attack in the rerouting scenario. The logic is shown in Figure 5.



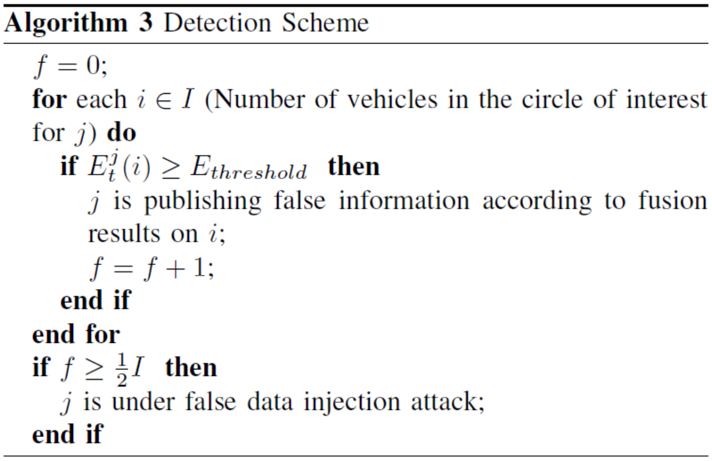
Algorithm 2 Decision Logic

```
for each $j \in N$(Number of neighboring vehicles) do
    if $E_t^j \geq E_{threshold}$ then
        $j$ is publishing false information;
    end if
end for
```

Figure 5: Decision logic for attack detection in rerouting scenario

In this logic, E_t^j shows the residue between estimated results after data fusion and information sent from neighboring vehicle j at time t. E_t^j is a vector that is as same as defined in Eq. (15). As false yaw angle attack is not considered, there are only three states for E_t^j and $E_{threshold}$. E_t^j exceeds $E_{threshold}$ if at least one element in the vector E_t^j exceeds the corresponding value in the matrix $E_{threshold}$.

A false data injection attack on that vehicle is detected when the residue is larger than a threshold ($E_{threshold}$). The thresholds are selected using the optimal threshold selection method by minimizing Eq. (18). As one neighboring vehicle j could also be the neighbors of other ego vehicles and there is a filter for each of these ego vehicles, there will be multiple results regarding one neighboring vehicle j. The majority rule is used here for consensus decision making as shown in Figure 6. If one neighboring vehicle j is identified as publishing false information by an ego vehicle, then a circle of interest for the vehicle j is created. All the ego vehicles around j in a sensing radius R which can report the decision result of j will be evaluated together. The total number of vehicles that report false data detected on j is defined as f. If more than half of the ego vehicles in the circle of interest report j is publishing false information, then j is considered as under false data injection attack.



Algorithm 3 Detection Scheme

```
$f = 0$;
for each $i \in I$ (Number of vehicles in the circle of interest
for $j$) do
    if $E_t^j(i) \geq E_{threshold}$ then
        $j$ is publishing false information according to fusion
        results on $i$;
        $f = f + 1$;
    end if
end for
if $f \geq \frac{1}{2}I$ then
    $j$ is under false data injection attack;
end if
```

Figure 6: Detection Scheme

If the vehicle j is considered as under attack for more than T seconds, the vehicle j is removed from the filter and stopped from being used in the data fusion algorithms. At this stage, the Cloud is still receiving information from vehicle j and evaluating the residue between information from j and the data fusion results from those ego vehicles in j's circle of interest. If more than half of the ego vehicles in the circle of interest report j is not publishing false information for T seconds, j will be put back in the data fusion algorithms.

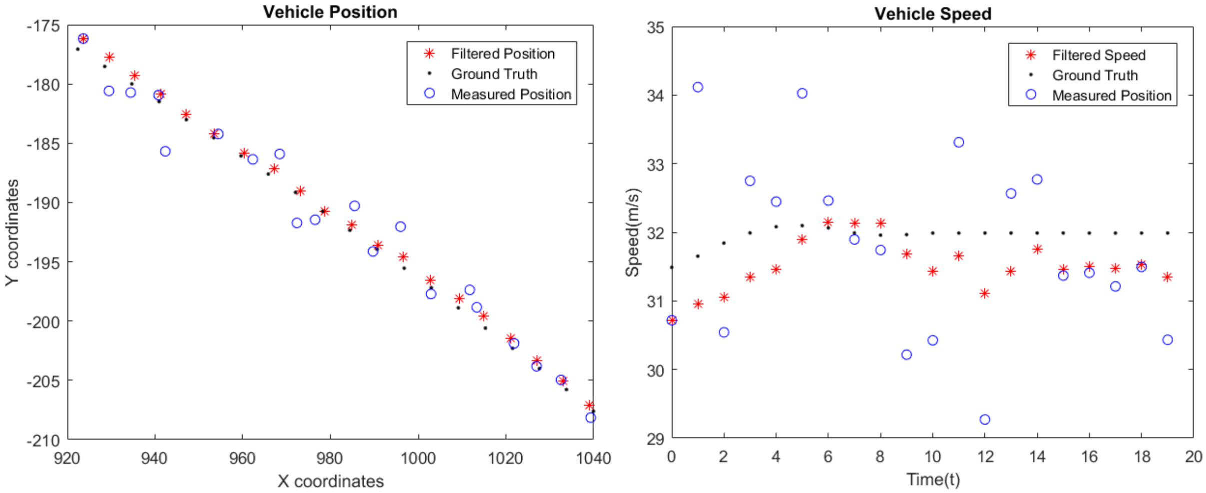
CHAPTER 4

Simulation Setup and Results

4.1 Platooning Scenario

The microscopic traffic simulation environment, VISSIM, generates vehicles with exponential interarrival times at the origin that traverse links based on the realistic vehicle following behavior. Interarrival times change as vehicles move along the network based on vehicle composition, vehicle characteristics, driving behavior, number of lanes, and other network settings. Similar to the complex real-life traffic systems with multiple parameters to control, a detailed analysis could be done in order to make sure a fully accurate comparison under different scenarios, which is beyond the scope of this report.

In this study, VISSIM simulations were controlled using the COM interface and MATLAB integration. For basic evaluation purposes, vehicles initialized at 53.00 kph are adopted within a 3-lane 2.37 km east and westbound links. A total of 30 platoons with 5 vehicles were generated. Initially, vehicles were generated from Normal distribution $N(10, 1)$ meters (m) apart (i.e., approximately 0.5 second (s) following time headways) and platoons generated 50 m apart from each other. Vehicles accelerate and decelerate based on platoon leader's speed information passed at 0.2 s intervals. Without attack, all platoons stay intact and stable. There are two types of false data injection attacks applied from 10 s to 12 s and from 15 s to 18 s in the simulation. The first one is a false position attack with 5 m added to the original coordinates and the second one is a false velocity attack with 4 m/s added to the original velocity. The connected vehicle market penetration rate was 60% for all runs. We selected this rate based on the experiments in Table 2, and more details are discussed later.



(a) Fusion results for position

(b) Fusion results for velocity

Figure 7: Data fusion results for an ego vehicle with ID 20

Figure 7 shows the estimated position and velocity for one ego vehicle on the road. Each data point represents the coordinates or velocity at a single timestamp. The Particle Filter based data fusion algorithm can filter noise in raw sensor data and gives a better estimation of vehicle states.

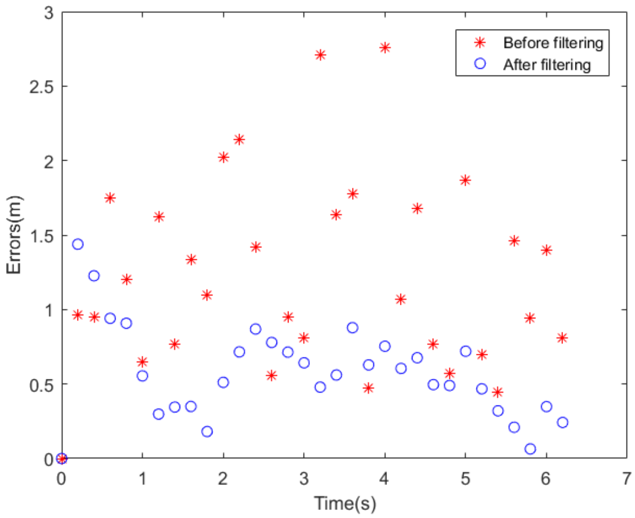
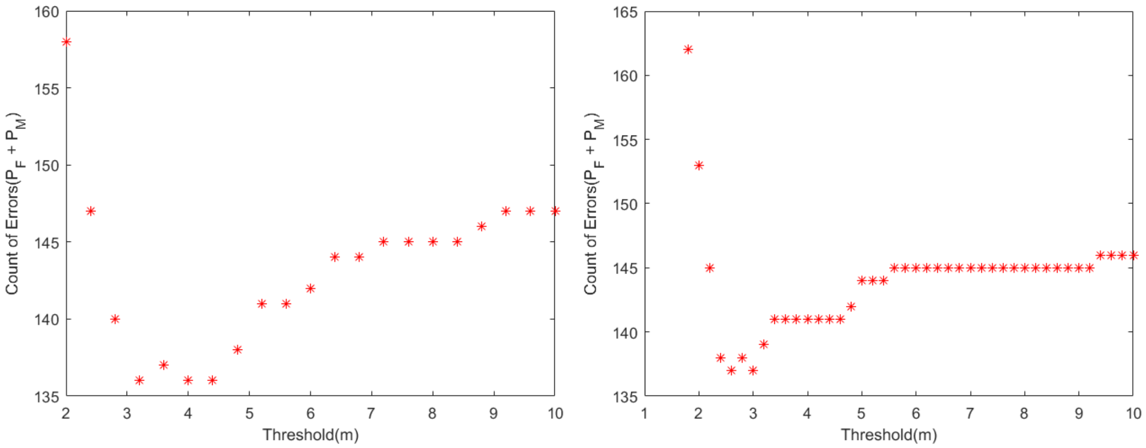


Figure 8: Error decreasing curve for an ego vehicle with ID 20

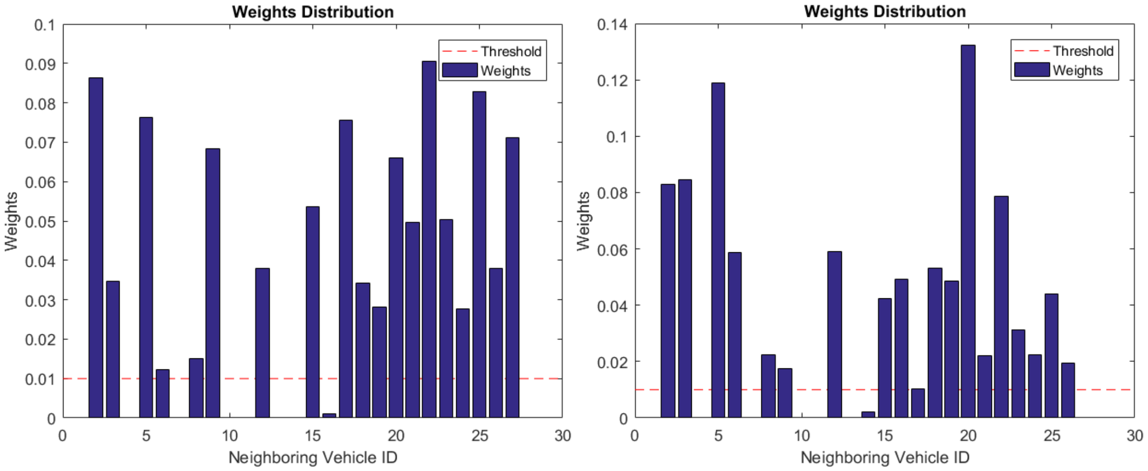
The error changing curve is shown in Figure 8, which demonstrates the decreasing trend. Notice that the error after filtering is lower than the original error.



(a) Residue threshold for position attack **(b) Residue threshold for velocity attack**

Figure 9: Residue threshold selection for false position and velocity attack

Figure 9 illustrates the optimal threshold selection result for this platooning scenario. The total probability of error P_E first decreases with the increase of threshold, and then P_E begins to increase after the threshold reaches 4.4 m and 3 m/s, respectively. Therefore, the threshold can be set to 4 m for false position attack and 2.8 m/s for false velocity attack. Here, the threshold depends on the minimum value of the false injected data to be detected and on the number of vehicles in the platoon that defines the distributions of the errors to be minimized in Eq. (21). The threshold is the minimal value that will be identified as an attack; thus, we only need to find the lower bound of the false attack range that can cause a degradation in the platooning scenario, and select the optimal threshold based on this lower bound. Then for any false data attacks beyond the lower bound, the threshold should be able to detect them.



(a) Weight threshold for position attack **(b) Weight threshold for velocity attack**

Figure 10: Weight threshold selection for false position and velocity attack

Figure 10 illustrates different weights for neighboring vehicles from the view of one ego vehicle ID 20. If the weight is lower than the threshold, then it is considered to have false data. In Table 2, we show the mean squared error (MSE) before filtering and after filtering for different CAV penetration rates. We can see from the table that the CAV penetration rate does not affect the MSE a lot and we suppose these benefits from the proposed two-stage architecture shown in Figure 2. As there are M regular particles that contribute directly to the filter outputs instead of directly using neighboring vehicles' estimation of ego vehicle states, the second stage of the proposed data fusion algorithm can be stable and eliminate some negative effects when there are not enough neighboring vehicles. In Table 2, we also showed the detection rate (true positive rate) and false alarm rate (false positive rate). Regarding these two matrices, the decrease in CAV penetration rate can also result in a decrease of false alarm rate while almost having no effect on the detection rate. Thus, we select 60% for all the other experiments shown before as this is the lowest penetration rate with a detection rate of 1.00.

Table 2: Effects of CAV Penetration Rates

CV Penetration Rate	MSE before filtering	MSE after filtering	Detection Rate	False Alarm Rate
100%	2.15	0.06	1.00	0.017
80%	2.19	0.05	1.00	0.017
60%	2.38	0.07	1.00	0.015
40%	2.22	0.07	0.96	0.013

4.2 Rerouting Scenario

The Cloud setup is based on Microsoft Azure and MATLAB as shown in Figure 12. A Linux virtual machine (VM) is created in Microsoft Azure with MATLAB installed on it. On the local machine, VISSIM is used to simulate the rerouting traffic scenario and VISSIM-MATLAB co-simulation is used to stream real-time traffic and vehicle data. In order to upload data to Cloud in real time, a UDP communication between two MATLAB sessions is established. The overall data flow acts as follows: first, the traffic scenario is simulated in VISSIM and relative vehicle data is streamed to MATLAB session 1; then, the data is transferred to MATLAB session 2 on Azure Cloud and used in the attack detection algorithm; finally, the results of attack detection unit are sent back to

MATLAB session 1 on the local machine for display. In this scenario, the local machine acts as an onboard embedded computer on vehicles, and the Azure Cloud acts as a kind of traffic management center.

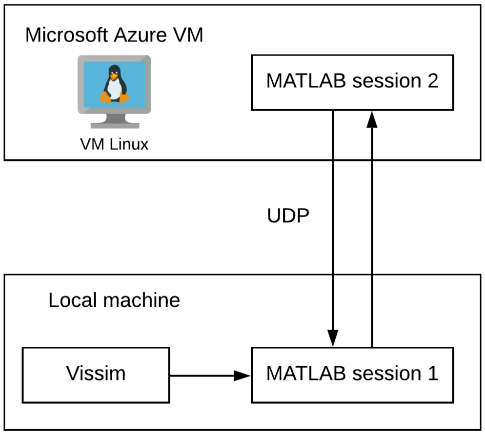


Figure 12: Overall architecture for Azure based simulation

The rerouting scenario is still built in VISSIM, and simulations were controlled using COM interface and MATLAB integration. Three links are created with the same origin and destination as shown in Figure 13. The links are single direction and single lane. The simulation step is 0.2 s and the routing algorithm is based on the one illustrated in section 3.3. Without attack, vehicles are able to select a suitable route with minimum travel cost. As the velocity of connected vehicles on road will affect travel time and thus affect the travel cost, false velocity attacks are performed in this scenario. Also, attacks on multiple vehicles are conducted as only one malicious velocity data on the road is not able to affect travel time, and travel cost a lot. More specifically, false velocity attacks are applied on 4 vehicles on the road where there are around 8 vehicles in total. The attacks are injected from 6 s to 12 s in the simulation and all the attacked vehicles have a malicious velocity which is 6 m/s lower than the original normal velocity. Therefore, fake congestion is created for that route. The connected vehicle market penetration rate was 90% for all runs. We selected a higher penetration rate in this rerouting scenario because a fake congestion attack needs at least a batch of connected vehicles to publish false slow speed values, which is different from the platooning scenario where a single false position or speed attack on the front car can cause degradation performance.

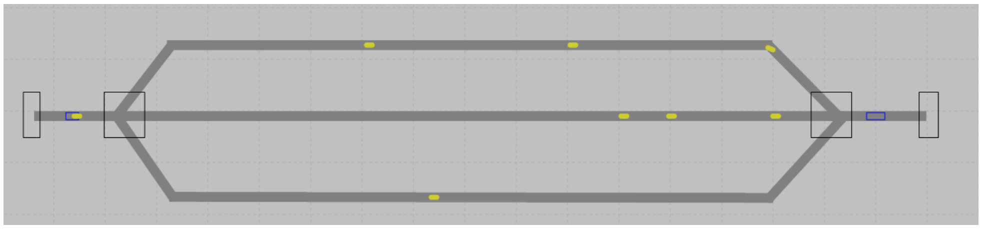
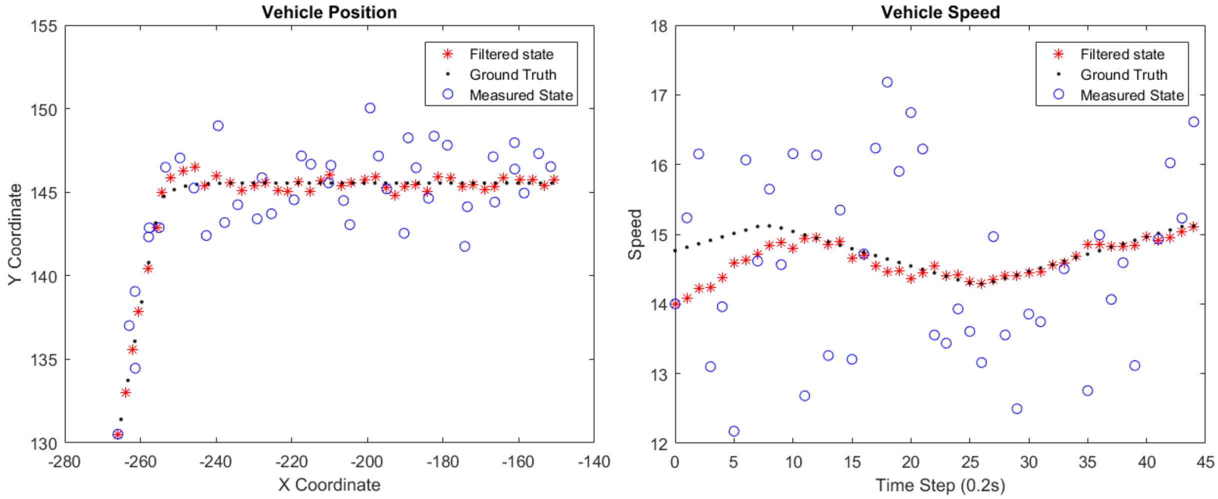


Figure 13: Rerouting Scenario in VISSIM

Figure 14 shows the estimated position and velocity for one ego vehicle on the road. Each data point represents the coordinates or velocity at a single time step. The Particle Filter based data fusion algorithm is able to filter noise in raw sensor data and gives a better estimation of vehicle states. The residue changing curve is shown in Figure 15, which demonstrates the decreasing trend. We can see the residue after filtering is lower than the original error. Figure 16 illustrates the optimal threshold selection result and the ROC curve for this rerouting scenario. The total probability of error P_E first decreases with the increase of threshold, and then P_E begins to increase after the threshold reaches 4 m/s. Therefore, the threshold can be set to 2 m/s for false velocity attacks. The selection of 2 m/s is suitable as it is close to the left upper corner of the figure and makes a balance of false alarm and misdetection.



(a) Fusion results for position **(b) Fusion results for velocity**

Figure 14: Data fusion results for an ego vehicle with ID 17

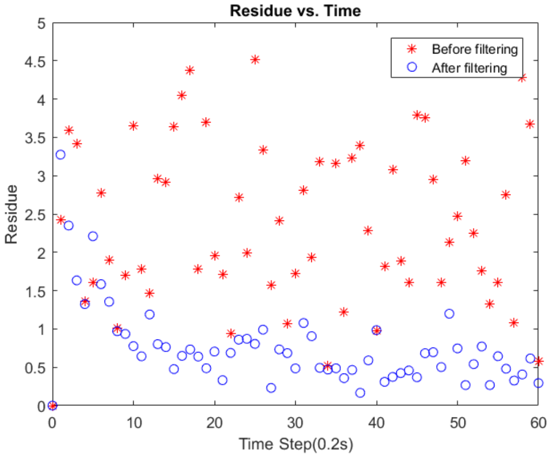
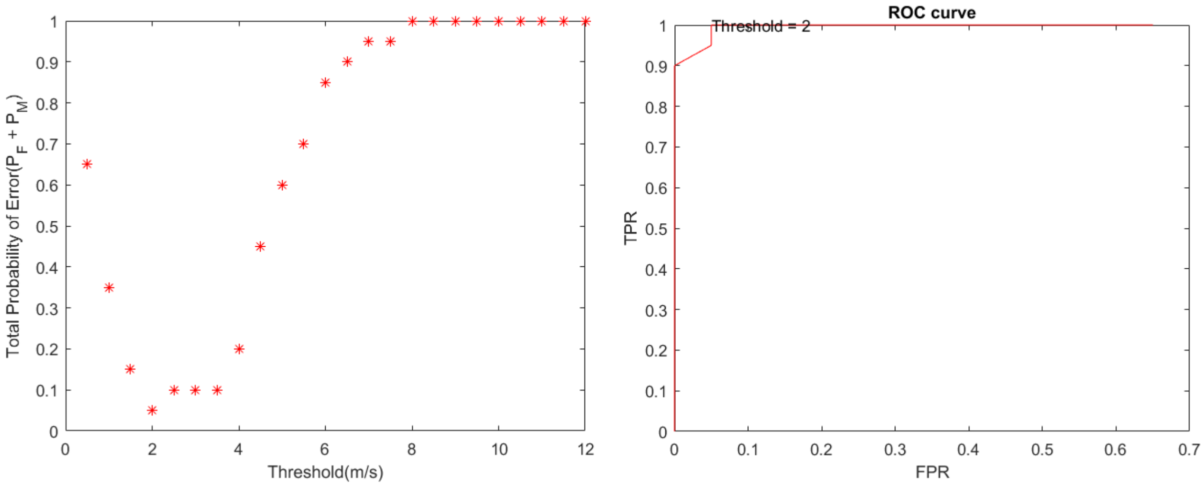


Figure 15: Residue decreasing curve for an ego vehicle with ID 17



(a) Optimal threshold selection **(b) ROC curve**

Figure 16: Residue threshold selection for false velocity attack

As we discussed before in section 4.1 for the platooning scenario, the threshold is relevant to the false injected data and is the minimal value to be identified as an attack, which means any values over the threshold should be able to be identified as an attack. Therefore, in this case, we only need to find the lower bound for the false injected data which can cause degradation performance (e.g., wrong routes selection). We leave a universal data range for false data injected attacks in transportation systems for future work, which needs a lot more work to analyze the actual performance under different scenarios and application settings.

As shown in Figure 17, the decision-making scheme is able to identify the attackers around 0.2 s after the attack is injected, which is the evaluation interval. Here vehicles ID'd 13, 14, 15, and 16 are identified as attackers. As the attacked vehicles will be removed from the fusion algorithm once they are detected, they won't affect the fusion results anymore. A comparison of the data fusion results without the attack mitigation and with the attack mitigation is shown in Figure 18. We could see that this mitigation approach can generate a flow of trustworthy information for CAVs, which can ensure the safety and correct behavior of connected vehicles.

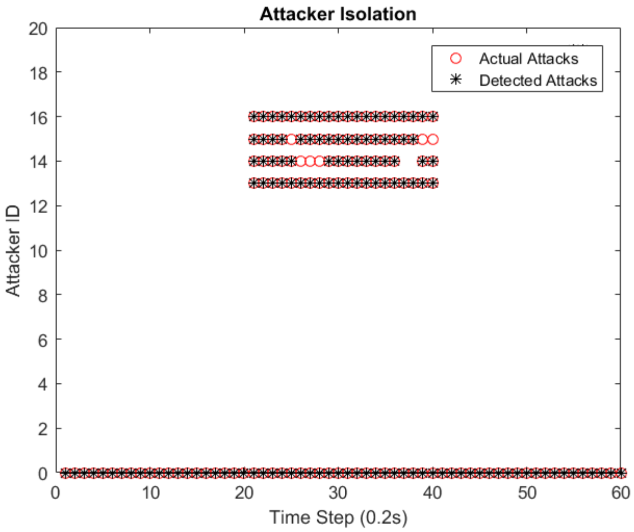
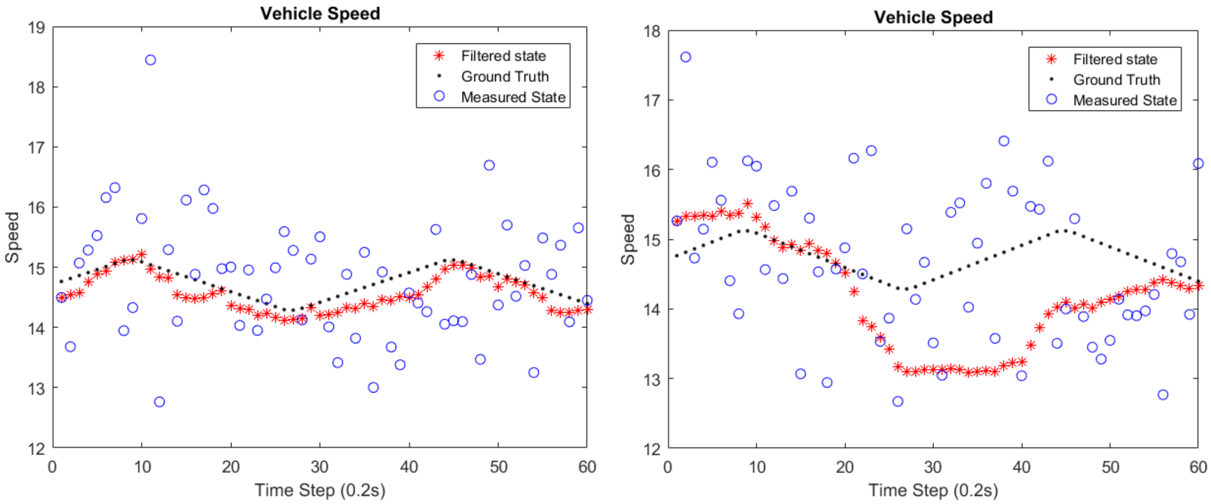


Figure 17: Attacker isolation results in rerouting scenario



(a) With attack mitigation **(b) Without attack mitigation**

Figure 18: Comparison of using attack mitigation and no attack mitigation

CHAPTER 5

Conclusions and Future Work

In this project, two different data fusion algorithms with different architecture and two attack detection decision schemes for connected vehicles are presented to mitigate false data injection attacks in CAV scenarios. The methodology leverages vehicles' connectivity and Particle Filters for vehicle state estimation and attack detection. The proposed approaches combine Particle Filters and vehicle-to-vehicle communication in order to fuse the location and speed information of multiple vehicles, then the results of the data fusion algorithm are used to construct the decision-making scheme in order to identify and isolate an attacker. The two decision schemes leverage the knowledge of diagnostics and consensus decision making. Two attack scenarios are modeled, which are a vehicle platooning scenario and a vehicle navigation routing scenario. The simulation results presented in this paper show the detection capability of the proposed approaches in real-time false data injection attacks on connected vehicles in the platooning and rerouting scenarios. The results demonstrate that cooperative localization can improve the location and speed estimation of vehicles, and the decision-making scheme is able to identify the vehicle which sends false information. More specifically, the results from the simulations are summarized as follows:

1. The localization accuracy for each CAV (i.e., both its location and speed) is able to be improved by the proposed data fusion techniques. Both the two-stage architecture and the multi-sensor architecture can reduce the error significantly.
2. The proposed two attack detection unit (ADU) inspired by the fault signature table and consensus decision making, are able to detect and isolate the attackers with a high detection rate (over 90%).
3. Cloud computing can be leveraged in CAV applications as it can provide powerful computing resources and can be even enhanced with 5G to provide huge data streaming.
4. The idea of cooperative localization can be used in accelerating the spreading of autonomous vehicles as V2V can work as complement sensors other than the onboard vehicle sensors. The adaption of V2V can provide redundant sensing information for autonomous vehicles to provide more robust perception results and can also mitigate potential cyber-attacks.

Possible future work includes modeling the communication delay in the network and predicting behaviors of unconnected vehicles. The modeling of the communication channels for CAVs is important for some safety-crucial scenarios and can make the simulations more realistic. Recently, the development of 5G technologies has shown an increasing trend in V2V communications. Also, some recent work (Balasubramanian et al. 2020) has investigated the integration of 5G with Cloud and Edge computing in the case of vehicular networks. It can be a strong improvement if the proposed method considers the latency, delay, or communication lost between vehicles and Clouds. In addition, the behavior prediction of unconnected vehicles also draws a lot of interest as the CAV generation rate cannot be guaranteed to be 100% all the time. The knowledge about unconnected vehicles in a mixed traffic scenario can help to identify the abnormal information existing in CAVs because unconnected vehicles are not threatened by cyber-attacks. The investigation of other types of attacks using the data fusion method could also be a future research direction. As we only consider the false position and velocity attacks in this paper, false yaw angle, yaw rate or acceleration attacks could also be investigated. More complex attack modeling can be investigated based on different traffic scenarios like lane changing behaviors or changing platooning average speed.

REFERENCES

Automotive v2x market by connectivity (dsrc, and cellular), communication (v2v, v2i, v2p, v2g, v2c, and v2d), vehicle (passenger car, and commercial vehicle), propulsion (ice and ev), unit, offering, technology, and region - global forecast to 2028. Markets and Markets. [Online]. https://www.marketsandmarkets.com/Market-Reports/automotive-vehicle-to-everything-v2x-market-90013236.html

Lu, N., Cheng, N., Zhang, N., Shen, X. and Mark, J.W., 2014. Connected vehicles: Solutions and challenges. IEEE internet of things journal, 1(4), pp.289-299.

Dey, K.C., Yan, L., Wang, X., Wang, Y., Shen, H., Chowdhury, M., Yu, L., Qiu, C. and Soundararaj, V., 2015. A review of communication, driver characteristics, and controls aspects of cooperative adaptive cruise control (CACC). IEEE Transactions on Intelligent Transportation Systems, 17(2), pp.491-509.

Kong, L., Khan, M.K., Wu, F., Chen, G. and Zeng, P., 2017. Millimeter-wave wireless communications for IoT-cloud supported autonomous vehicles: Overview, design, and challenges. IEEE Communications Magazine, 55(1), pp.62-68.

Fallah, Y.P., Huang, C., Sengupta, R. and Krishnan, H., 2010, April. Design of cooperative vehicle safety systems based on tight coupling of communication, computing and physical vehicle dynamics. In Proceedings of the 1st ACM/IEEE international conference on cyber-physical systems (pp. 159-167).

Work, D.B. and Bayen, A.M., 2008, November. Impacts of the mobile internet on transportation cyberphysical systems: traffic monitoring using smartphones. In National Workshop for Research on High-Confidence Transportation Cyber-Physical Systems: Automotive, Aviation, & Rail (pp. 18-20).

Petit, J. and Shladover, S.E., 2014. Potential cyberattacks on automated vehicles. IEEE Transactions on Intelligent transportation systems, 16(2), pp.546-556.

Chowdhury, M., Islam, M. and Khan, Z., 2019. Security of Connected and Automated Vehicles. The Bridge, 49(3).

P., Liu, H., Wang, H. and Zhang, S., 2017, April. Securing wireless communications of connected vehicles with artificial intelligence. In 2017 IEEE international symposium on technologies for homeland security (HST) (pp. 1-7). IEEE.

Singh, M., Singh, D. and Jara, A., 2015, October. Secure cloud networks for connected & automated vehicles. In 2015 International Conference on Connected Vehicles and Expo (ICCVE) (pp. 330-335). IEEE.

Calvo, J.A.L. and Mathar, R., 2018, June. Secure blockchain-based communication scheme for connected vehicles. In 2018 European Conference on Networks and Communications (EuCNC) (pp. 347-351). IEEE.

Alheeti, K.M.A. and McDonald-Maier, K., 2016, September. Hybrid intrusion detection in connected self-driving vehicles. In 2016 22nd International Conference on Automation and Computing (ICAC) (pp. 456-461). IEEE.

Li, W. and Song, H., 2015. ART: An attack-resistant trust management scheme for securing vehicular ad hoc networks. IEEE Transactions on Intelligent Transportation Systems, 17(4), pp.960-969.

Liang, J., Lin, Q., Chen, J. and Zhu, Y., 2019. A filter model based on hidden generalized mixture transition distribution model for intrusion detection system in vehicle ad hoc networks. IEEE Transactions on Intelligent Transportation Systems.

Sedjelmaci, H., Senouci, S.M. and Ansari, N., 2016. Intrusion detection and ejection framework against lethal attacks in UAV-aided networks: A Bayesian game-theoretic methodology. IEEE Transactions on Intelligent Transportation Systems, 18(5), pp.1143-1153.

Cardenas, A., Amin, S., Sinopoli, B., Giani, A., Perrig, A. and Sastry, S., 2009, July. Challenges for securing cyber physical systems. In Workshop on future directions in cyber-physical systems security (Vol. 5, No. 1).

Biron, Z.A., Dey, S. and Pisu, P., 2017, May. Resilient control strategy under denial of service in connected vehicles. In 2017 American Control Conference (ACC) (pp. 4971-4976). IEEE.

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

Rayamajhi, A., Biron, Z.A., Merco, R., Pisu, P., Westall, J.M. and Martin, J., 2018, May. The impact of dedicated short range communication on cooperative adaptive cruise control. In 2018 IEEE International Conference on Communications (ICC) (pp. 1-7). IEEE.

van Wyk, F., Wang, Y., Khojandi, A. and Masoud, N., 2019. Real-time sensor anomaly detection and identification in automated vehicles. IEEE Transactions on Intelligent Transportation Systems, 21(3), pp.1264-1276.

Petrillo, A., Pescape, A. and Santini, S., 2017, June. A collaborative control strategy for platoons of autonomous vehicles in the presence of message falsification attacks. In 2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS) (pp. 110-115). IEEE.

Fiengo, G., Petrillo, A., Salvi, A., Santini, S. and Tufo, M., 2016, December. A control strategy for reducing traffic waves in delayed vehicular networks. In 2016 IEEE 55th Conference on Decision and Control (CDC) (pp. 2462-2467). IEEE.

Zhang, L. and Orosz, G., 2016. Motif-based design for connected vehicle systems in presence of heterogeneous connectivity structures and time delays. IEEE Transactions on Intelligent Transportation Systems, 17(6), pp.1638-1651.

Mousavinejad, E., Yang, F., Han, Q.L., Ge, X. and Vlacic, L., 2019. Distributed cyber attacks detection and recovery mechanism for vehicle platooning. IEEE Transactions on Intelligent Transportation Systems.

Kong, S.H. and Jun, S.Y., 2017. Cooperative positioning technique with decentralized malicious vehicle detection. IEEE Transactions on Intelligent Transportation Systems, 19(3), pp.826-838.

Heng, L., Work, D.B. and Gao, G.X., 2014. GPS signal authentication from cooperative peers. IEEE Transactions on Intelligent Transportation Systems, 16(4), pp.1794-1805.

Bak, S., Manamcheri, K., Mitra, S. and Caccamo, M., 2011, April. Sandboxing controllers for cyber-physical systems. In 2011 IEEE/ACM Second International Conference on Cyber-Physical Systems (pp. 3-12). IEEE.

Sha, L., 2001. Using simplicity to control complexity. IEEE Software, 18(4), pp.20-28.

Golestan, K., Sattar, F., Karray, F., Kamel, M. and Seifzadeh, S., 2015. Localization in vehicular ad hoc networks using data fusion and V2V communication. Computer Communications, 71, pp.61-72.

Zeng, K.C., Shu, Y., Liu, S., Dou, Y. and Yang, Y., 2017, February. A practical GPS location spoofing attack in road navigation scenario. In Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications (pp. 85-90).

Lin, J., Yu, W., Zhang, N., Yang, X. and Ge, L., 2018. Data integrity attacks against dynamic route guidance in transportation-based cyber-physical systems: Modeling, analysis, and defense. IEEE Transactions on Vehicular Technology, 67(9), pp.8738-8753.

Coles, C., 2019. 11 advantages of cloud computing and how your business can benefit from them. McAfee. [Online]. https://www.skyhighnetworks.com/cloud-security-blog/11-advantages-of-cloud-computing-and-how-your-business-can-benefit-from-them/

12 benefits of cloud computing. Salesforce. [Online]. https://www.salesforce.com/products/platform/best-practices/benefits-of-cloud-computing/

Palmer, M., The truth about virtual machines in the cloud, 2018. [Online]. https://vmiss.net/2018/10/12/virtual-machine-cloud-computing/

Tian, D., Yuan, Y., Qi, H., Lu, Y., Wang, Y., Xia, H. and He, A., 2015. A dynamic travel time estimation model based on connected vehicles. Mathematical Problems in Engineering, 2015.

Deng, H.W., Rahman, M., Chowdhury, M., Salek, M.S. and Shue, M., 2020. Commercial Cloud Computing for Connected Vehicle Applications in Transportation Cyber-Physical Systems. arXiv preprint arXiv:2008.07290.

Dey, K.C., Yan, L., Wang, X., Wang, Y., Shen, H., Chowdhury, M., Yu, L., Qiu, C. and Soundararaj, V., 2015. A review of communication, driver characteristics, and controls aspects of cooperative adaptive cruise control (CACC). IEEE Transactions on Intelligent Transportation Systems, 17(2), pp.491-509.

Hussain, R., Rezaeifar, Z. and Oh, H., 2015. A paradigm shift from vehicular ad hoc networks to VANET-based clouds. Wireless Personal Communications, 83(2), pp.1131-1158.

Nasimi, M., Habibi, M.A. and Schotten, H.D., 2020. Platoon--assisted Vehicular Cloud in VANET: Vision and Challenges. arXiv preprint arXiv:2008.10928.

Montanaro, U., Fallah, S., Dianati, M., Oxtoby, D., Mizutani, T. and Mouzakitis, A., 2018, October. On a fully self-organizing vehicle platooning supported by cloud computing. In 2018 Fifth International Conference on Internet of Things: Systems, Management and Security (pp. 295-302). IEEE.

Chang, B.J., Tsai, Y.L. and Liang, Y.H., 2017. Platoon-based cooperative adaptive cruise control for achieving active safe driving through mobile vehicular cloud computing. Wireless Personal Communications, 97(4), pp.5455-5481.

Balasubramanian, V., Otoum, S., Aloqaily, M., Al Ridhawi, I. and Jararweh, Y., 2020. Low-latency vehicular edge: A vehicular infrastructure model for 5G. Simulation Modelling Practice and Theory, 98, p.101968.