University of Cyprus
Department of Electrical and Computer Engineering

DETECTING AND COUNTERING CYBER-ATTACKS IN CAVS

Syeda Zillay Nain Zukhraf

Submitted to the University of Cyprus in partial fulfillment of the requirements

of the Master in Science (MSc) degree in *"Intelligent Critical Infrastructure Systems"*

Master of Science in
INTELLIGENT CRITICAL
INFRASTRUCTURE SYSTEMS

Department of Electrical and Computer Engineering

University of Cyprus

June 2023

DETECTING AND COUNTERING CYBER-ATTACKS IN CAVS

Syeda Zillay Nain Zukhraf

Thesis Examination Committee

Prof. Maria Michael, Co-Advisor, Chair of the Committee, ECE-Department

Prof. George Ellinas, Committee Member, ECE-Department

Prof. Theocharis Theocharides, Committee Member, ECE-Department

Dr. Christos Laoudias, Co-advisor, KIOS Centre of Excellence

Abstract

Connected and Autonomous Vehicles (CAVs) are becoming increasingly popular due to their ability to make driving safer and more efficient. However, these vehicles rely heavily on GPS systems to determine their location and route. Unfortunately, GPS systems are vulnerable to spoofing attacks, where malicious actors can provide false location information to the CAV, potentially causing accidents or other dangerous situations.

In the literature foreseen, there is a limitation for the detection and mitigation technique for GPS spoofing. To address this problem, we have developed a solution by using in-vehicle detection. The vehicle takes information from the sensors of the car like Inertial Measurement Unit (IMU), Global Navigation Satellite System (GNSS), and Odometer, and detects any anomalies that may indicate a spoofing attack. This approach is practical and effective, as it does not require information from neighboring vehicles, which can be unreliable in certain situations.

Another analysis as a part of this thesis is by utilizing Machine Learning (ML) algorithms. The traditional attack detection solutions developed by using ML techniques require normal and attack data labels. Obtaining 'normal' and 'attack' data labels in practical or controlled settings is challenging for conventional attack detection methods. To address this limitation, we are utilizing the ML algorithm only to process attack-free scenarios for the training stage. Different ML techniques are used that support the detection of anomalies. This proposed approach is effective as it does not require labeled data and can adapt to new and evolving attack strategies.

The suggested solution builds on a previous research project at KIOS, called the " CARAMEL-In-vehicle Detection Solution ". By combining In-vehicle Detection Solution and machine learning-based techniques, this solution can effectively detect GPS spoofing attacks in CAVs.

Acknowledgments

I would like to convey my deepest gratitude and appreciation to the following persons for their contributions to the completion of my thesis:

First and foremost, I would like to express my heartfelt gratitude to my research supervisors, Dr. Christos Laoudias and Prof. Maria K. Michael, for their important advice, insightful input, and continuous support during my studies. Their knowledge, insight, and devotion have been invaluable in defining my research and guiding me through the complexity of my study. I am extremely appreciative for their guidance and the chances they have provided.

I would also want to thank Demetrios Hadjizorzis and Nicolas Souli from the KIOS Lab for their help and support during my thesis. Their contributions, which included their knowledge, technical expertise, and insightful conversations, considerably enhanced my work. I like their ongoing availability, collaborative energy, and readiness to share their knowledge.

I would want to express my profound gratitude to the University of Patras for their assistance and participation in expanding my thesis work, as mentioned in Publication Section. Their knowledge and assistance were critical in expanding the breadth and depth of my research. As noted in Publication Section, I would also want to thank the machine learning team especially Stylianos Filippou, Andreas Achilleos and Kleanthis Malialis at the KIOS Research and Innovation Center of Excellence (KIOS CoE) for their substantial contribution and advice. Their knowledge of machine learning techniques has tremendously expanded my study and offered useful ideas for future improvement.

Furthermore, I would want to thank everyone who has contributed to my study, either directly or indirectly, by sharing their knowledge, expertise, and experiences. Your assistance, advice, and contributions have been priceless, and I am thankful for everything you have done.

Finally, I'd want to thank the academic and administrative personnel at KIOS CoE and UCY for creating a welcoming learning atmosphere and giving access to resources that have aided my research efforts.

Most importantly I want to express my heartfelt appreciation to my family for their consistent support and encouragement during my Master's program. My success has been driven by their love, understanding, and belief in my ability. This thesis is dedicated to my father Nuh Shoaib, my mother Raheela, my sister Ghanwa, and my fiancé Ammar Mehdi.

Ευχαριστίες

Θα ήθελα να εκφράσω τις βαθύτερες μου ευγνωμοσύνες και εκτίμηση προς τα παρακάτω άτομα για τη συνεισφορά τους στην ολοκλήρωση της διατριβής μου:

Καταρχάς, θέλω να εκφράσω την ευγνωμοσύνη μου προς τους επιβλέποντές μου, τον Δρ. Χρίστος Λαουδιάς και την Καθηγήτρια Μαρία Κ. Μιχαήλ, για τις σημαντικές συμβουλές και τη συνεχή υποστήριξή τους κατά τη διάρκεια των σπουδών μου. Η γνώση, η εμπειρία και η αφοσίωσή τους ήταν ανεκτίμητες για τον καθορισμό της έρευνάς μου και την καθοδήγησή μου μέσα στην πολυπλοκότητα της μελέτης μου. Είμαι εξαιρετικά ευγνώμον για την καθοδήγησή τους και τις ευκαιρίες που μου παρείχαν.

Θα ήθελα επίσης να ευχαριστήσω τον Δημήτριος Χατζηζώρζης και τον Νικόλας Σούλη για τη βοήθεια και την υποστήριξή τους κατά τη διάρκεια της διατριβής μου. Η συνεισφορά τους, που περιλάμβανε τη γνώση τους, την τεχνική τους εμπειρία και τις ενδιαφέρουσες συζητήσεις, ενισχύσαν σημαντικά την εργασία μου. Εκτιμώ τη συνεχή τους διαθεσιμότητα και την προθυμία τους να μοιραστούν τις γνώσεις τους.

Θα ήθελα να εκφράσω τη βαθιά μου ευγνωμοσύνη προς το Πανεπιστήμιο Πατρών για την υποστήριξη και τη συμμετοχή τους στη διεύρυνση της διατριβής μου, όπως αναφέρεται στην Ενότητα Δημοσιεύσεων. Η γνώση και η υποστήριξή τους ήταν κρίσιμες για τη διεύρυνση της έρευνάς μου σε όλο το φάσμα της. Όπως αναφέρεται επίσης στην Ενότητα Δημοσιεύσεων, θα ήθελα επίσης να ευχαριστήσω την ομάδα μηχανικής μάθησης, ιδιαίτερα τον Στυλιανός Φιλίππου, τον Ανδρέας Αχιλλέως και τον Κλεάνθης Μαλιαλής, στο Κέντρο Αριστείας Έρευνας και Καινοτομίας KIOS, για την σημαντική τους συνεισφορά και συμβουλές. Η γνώση τους στις τεχνικές της μηχανικής μάθησης επέκτεινε σημαντικά τη μελέτη μου και πρόσφερε χρήσιμες ιδέες για μελλοντικές βελτιώσεις.

Επιπλέον, θα ήθελα να ευχαριστήσω όλους όσους συνέβαλαν στη μελέτη μου, είτε άμεσα είτε έμμεσα, μοιράζοντας τις γνώσεις, την εμπειρία και τις εμπειρίες τους. Η βοήθειά σας, οι συμβουλές σας και οι συνεισφορές σας ήταν ανεκτίμητες, και είμαι ευγνώμον για όλα όσα έχετε κάνει.

Τέλος, θα ήθελα να ευχαριστήσω το ακαδημαϊκό και διοικητικό προσωπικό του Κέντρου Αριστείας Έρευνας και Καινοτομίας KIOS και του Πανεπιστημίου Κύπρου για τη δημιουργία μιας φιλόξενης ατμόσφαιρας μάθησης και την πρόσβαση σε πόρους που βοήθησαν τις προσπάθειες της έρευνάς μου.

Τέλος, θέλω να εκφράσω την ιδιαίτερη ευγνωμοσύνη μου στην οικογένειά μου για τη συνεχή υποστήριξη και ενθάρρυνση κατά τη διάρκεια του μεταπτυχιακού μου προγράμματος. Η επιτυχία μου οφείλεται στην αγάπη, την κατανόηση και την πίστη τους στις ικανότητές μου. Αυτή η διατριβή αφιερώνεται στον πατέρα μου Nuh Shoaib, τη μητέρα μου Raheela, την αδελφή μου Ghanwa και τον αρραβωνιαστικό μου Ammar Mehdi.

Publications

C1. S. N. Piperigkos, A. S. Lalos, C. Anagnostopoulos, **S. Z. N. Zukhraf**, C. Laoudias, and M. K. Michael, “Robust Cooperative Sparse Representation Solutions for Detecting and Mitigating Spoofing Attacks in Autonomous Vehicles." at MED The 31st Mediterranean Conference on Control and Automation. Limassol, Cyprus. 2023 Page 1-6.[Accepted]

C2. S. Filippou, A. Achilleos, **S. Z. N. Zukhraf**, C. Laoudias, K. Malialis, M. K. Michael and G. Ellinas, “A Machine Learning Approach for Detecting GPS Location Spoofing Attacks in Autonomous Vehicles." at VTC The 97th Vehicular Technology Conference. Florence, Italy. 2023 Page 1-7.[Accepted]

Table of Contents

List of Figures

List of Tables

Chapter 1 - Introduction

1.1 Background

By 2050, the CAVs industry is projected to achieve a market value of $7 trillion [1]. This growth has been accompanied by the implementation of updated standards for vehicular communications such as IEEE 802.11p and LTE-PC5, which operate without reliance on infrastructure and use unlicensed frequency bands. However, these advancements in connectivity have also brought about a significant rise in the potential for cyber-threats [2].

CAVs become more prevalent, there is a genuine concern regarding the possibility of cyber-attacks targeting these vehicles by exploiting various vulnerabilities. Since CAVs heavily rely on digital connectivity and communication systems, they become potential targets for hackers who may exploit weaknesses in their software, hardware, or network infrastructure. These cyber-attacks could encompass unauthorized access to the vehicle's systems, manipulation of critical functions, theft of sensitive data, or even remote control of the vehicle. The increasing connectivity of CAVs amplifies the risk of cyber threats, emphasizing the importance of implementing robust security measures to safeguard these vehicles and ensure the safety of passengers and the general public. The automotive industry is increasingly concerned about attacks targeting the GPS receiver of CAVs, particularly spoofing attacks on GPS locations. This kind of attack poses a significant threat as the security of CAVs can be compromised, potentially resulting in severe consequences for both drivers and pedestrians. Currently, there are detection solutions available that rely on specific hardware like antenna arrays and satellite signal processing techniques, which offer high accuracy but they are expensive in terms of installation in the CAVs and are bulky too. Therefore, there is a growing need for lightweight and cost-effective solutions that can effectively detect GPS spoofing attacks in order to address this issue.

The dependable and secure functioning of GPS sensors plays a vital role in ensuring the broader adoption of CAVs and the implementation of Vehicular Ad-hoc Networks (VANETs). GPS sensors are essential for both as they provide accurate positioning and navigation information, enabling various critical functionalities. Reliable GPS functioning is critical for CAVs performing activities such as autonomous driving, route planning, and collision avoidance. Likewise, VANETs rely largely on GPS sensors to enable vehicle communication and coordination for traffic management, congestion control, and safety applications. To improve the acceptance and efficacy of CAVs and VANETs in real-world circumstances, it is critical to ensure the dependability and security of GPS sensors [3]. GPS readings and detailed maps are critical for autonomous cars to operate effectively and autonomously. Vehicles may calculate

the most direct and efficient route from one site to another by using GPS data and high-definition maps. The lack of human intervention in the navigation process is eliminated by this autonomy. Vehicles optimize their functioning and save journey time by picking the quickest way independently. This level of autonomy dramatically improves transportation networks ease and efficacy by allowing cars to function effectively and autonomously [4].

1.2 Problem Statement

GPS is vulnerable to attacks such as spoofing and jamming. Jamming involves transmitting disruptive signals on the same frequencies as GPS signals, completely blocking its operation. Spoofing, on the other hand, deceives users by transmitting signals that mimic legitimate GPS satellite signals. GPS location spoofing poses a critical threat to the safety of CAV users, with various open-source resources available for carrying out such attacks. To address these risks, ongoing efforts focus on securing sensor systems and utilizing commercially available receivers to study vulnerabilities and potential threats.

1.3 Motivation

GPS spoofing exploits the stable characteristics of GPS signals, allowing attackers to generate fake signals that resemble genuine ones, leading to incorrect position, velocity, and time solutions for GPS receivers [5-7]. Advanced attacking strategies typically require patience on the part of the attacker. To carry out a successful attack on a GPS receiver, the attacker must synchronize their disruptive signals with the satellite signals. By increasing the power of their signals, attacker can manipulate the GPS to lock onto the false signals, thus exerting control over the victim's location. Several techniques exist for GPS spoofing, including Lift-off-aligned, Lift-off-delay, meaconing, jam and spoof, and trajectory spoofing. In terms of defense, various techniques focus on monitoring the signal power, analyzing the characteristics of signal at arrival, examining correlation of signal peaks, and fusion of multi-sensor employment. The cooperative localization approach in VANETs aligns with concept of collaborative defense, where network nodes exchange measurements to improve accuracy of location, detect spoofing attacks on GPS and mitigate them. [8-13].

To detect GPS spoofing attacks, current solutions rely on either data-driven approaches [14-17] or signal processing techniques [18-19]. Large volumes of data are analyzed and processed in data-driven ways to find trends and abnormalities that may signal a spoofing assault. Signal processing techniques, on the other hand, concentrate on studying the properties of GPS signals to detect any symptoms of faking. These methods use a variety of algorithms and strategies to distinguish between genuine GPS signals and spoof signals.

1.4 Contribution

Researchers and practitioners want to create effective solutions for detecting and combating GPS spoofing attacks by utilizing data-driven and signal processing technologies.

Several ML methods have been developed to identify irregularities related with GPS spoofing attempts. These systems use artificial intelligence to evaluate trends, learn from prior data, and spot variations that may indicate spoofing. [20-26].

This thesis builds upon the research conducted in papers [14] and [27] to introduce and assess a framework designed for detecting GPS location spoofing attacks within a vehicle. The design of the system suggested in this study makes use of data fusion from numerous sources. The following are the important contributions of this thesis:

1. GPS spoofing detection method in-vehicle: This study presents a viable and effective approach for detecting and mitigating GPS spoofing attacks within the vehicle itself. The car can detect irregularities that suggest a possible spoofing attack by analyzing data from multiple sensors such as the IMU, GNSS, and Odometer. This method is useful since it does not rely on information from nearby cars, which might be untrustworthy in some scenarios.
2. ML techniques for attack detection: For normal and attack settings, traditional ML-based attack detection algorithms require labeled data. Obtaining such labeled data in realistic or controlled circumstances, on the other hand, might be difficult. To overcome this limitation, this research employs ML algorithms solely for processing attack-free scenarios during the training stage. Different ML techniques are utilized to detect anomalies, enabling the system to adapt to new and evolving attack strategies. This approach is effective as it does not rely on labeled data and can enhance the detection capabilities of the system.

The rest of the thesis is structured as follow. Chapter 2 overviews the related work on GPS location spoofing attack detection methods. Attack detection frame work different sensors calibrations and attack generation model is described in Chapter 3. A part form this Chapter 3 also contains the extended work to this thesis which was done in collaboration of University of Patras Team in Section 3.8. Chapter 4 presents the frame work of Machine learning and deep learning models-based solution and their comparative results. At last Chapter 5 consists of the Conclusion and Future works.

Chapter 2 - Related Work

Signal processing-based solutions often rely on specialized equipment to detect and mitigate GPS spoofing attacks. The research [18] displays real-time GPS spoofing detection utilizing software defined radios (SDRs) as one example. The raw (I,Q) coefficients from the radio frequency front end are captured by SDRs. The phase difference of GPS signals obtained from separate antennas is then used to calculate these coefficients. The system can determine the position and identify possible spoofing by comparing the signals and measuring their interference levels. The study makes use of an open-source framework called GNSS-SDR to help with signal processing and analysis. In the mentioned study [19], vehicles use dedicated short-range communication to share GPS code pseudo-range measurements with other vehicles. Each car performs a linear operation on the GPS data received, generating separate statistics based on the measurements of each nearby vehicle. These statistics are then utilized to conduct a cumulative summing operation at each car, with the goal of detecting a strong correlation of arrival times for any faked GPS signals. The vehicles communicate their local detection values to a predetermined head vehicle. The head vehicles use a minimum-maximum change detection approach to maximize worldwide detection of GPS spoofing. The system improves its capacity to detect and neutralize spoofing attempts throughout the network of cars by merging local detection values and employing this optimization approach.

Data-driven solutions for detecting GPS location spoofing attacks utilize data as input and apply specific algorithms to mitigate the adversarial activity. In one particular solution found in the literature, in-vehicle multisensory data (such as accelerometer, gyroscope, compass, etc.) are leveraged to compute a parallel GPS-free stream of estimated vehicle locations. This is achieved through a fallback localization method based on Bayesian filtering. The solution utilizes the available sensor data to estimate the vehicles' positions, providing an alternative source of location information independent of GPS signals. By employing Bayesian filtering techniques, the system can enhance the accuracy and reliability of the estimated vehicle locations, contributing to the detection and mitigation of GPS location spoofing attacks. By comparing the estimated vehicle position with the GPS location reading, this facilitates the identification of possible location spoofing attacks [14]. The second method presents a collective defensive strategy against GPS spoofing attacks, which is especially useful within a VANET. This approach employs multi-modal sensor fusion, which combines data from many sensors, to improve the detection and mitigation of spoofing attacks. This fusion method incorporates metrics like as relative distances, relative angles, and relative azimuth angles among the vehicles in the network. Furthermore, the absolute location measurements of every vehicle received by GPS positions are considered. The collaborative defensive mechanism attempts to increase the

accuracy and reliability of detecting GPS spoofing attacks within the VANET environment by aggregating and analyzing this broad range of sensor data. [15] Several strategies for detecting GPS spoofing have been presented in the available literature. One prominent method, detailed in [16], compares accelerometer data from the vehicle to predicted acceleration measurements from the GPS device. A substantial discrepancy that exceeds a predefined threshold indicates the presence of a GPS spoofing assault. The decision variable is calculated using the acceleration error matrix, which quantifies the difference between the actual and estimated values. The threshold is set based on the desired likelihood of false alarm, establishing a balance between detecting spoofing attacks accurately and avoiding false positive outcomes. This technique uses accelerometer data and acceleration measurement comparison to give an extra layer of security against GPS spoofing attempts. Researchers have looked into using multi-sensor fusion (MSF) approaches to predict the vehicle's position in order to reduce reliance on GPS signals totally [17]. To improve the accuracy and resilience of the predicted vehicle location, these algorithms combine input from many sensors, including as cameras, LiDAR, radar, and inertial sensors. The MSF model reduces the probability of off-road and wrong-way attacks in autonomous cars by combining information from several sources. The threat model utilized in the technique considers an attacker's possible capabilities as well as assumptions about the CAV's control mechanisms. By examining a broader variety of sensor data and various attack scenarios, this comprehensive methodology attempts to improve the security and dependability of the CAVs positioning system.

ML methods have been used to detect location spoofing attacks as the amount of sensory data generated by vehicles has increased. However, there is a scarcity of study in this field that is primarily focused on autonomous vehicles. As a result, it is critical to investigate existing studies in related domains, such as autonomous aerial vehicles, to gain insights and comprehend how ML methods can be applied. ML methods can be categorized broadly into two different groups: supervised methods and unsupervised methods. Supervised methods necessitate ground truth information, which involves labeling data as either normal or indicating an attack in the scenario of location spoofing. These methods use labeled data to train models and predict whether a new data point represents a legitimate or spoofed location. Unsupervised methods, on the other hand, do not rely on pre-labeled data. Instead, they seek to detect patterns or abnormalities in data without being aware of individual assault occurrences. Even in the absence of labeled data, these algorithms use the intrinsic properties and distributions of the data to detect probable location spoofing attempts. Both supervised and unsupervised machine learning approaches have advantages and disadvantages and can be used depending on the availability of labeled data and the desired detection capabilities. Each solution delivers distinct insights

and helps to the broader understanding of identifying location spoofing attempts using ML methods.

The authors of reference [20] use two specific algorithms in the overall framework of supervised machine learning (ML) methods for detecting location spoofing attacks: k-Nearest Neighbor (k-NN) and Support Vector Machine (SVM). To classify instances based on their similarity to neighboring data points, the k-Nearest Neighbor algorithm is used. This algorithm considers features such as location and movement plausibility checks when detecting location spoofing. The algorithm determines whether a given instance is normal or indicative of location spoofing by comparing its characteristics with those of its k nearest neighbors in the training data. SVM is a classification approach that seeks to identify an ideal hyperplane to divide distinct classes of data points. SVM is trained to detect location spoofing using labeled data, where instances are classified as either normal or spoofed. The approach learns the patterns and boundaries that differentiate these classes, allowing it to categorize new instances as normal or indicative of location spoofing. In the referenced work [21], To detect GPS spoofing signals on Unmanned Aerial Vehicles (UAVs), an artificial Neural Network (NN) is used as a detection technique. Various information is acquired from the incoming GPS signals, which act as input features, to train and use the NN. The number of satellites, which refers to the number of GPS satellites from which the UAV gets signals, is one of these input characteristics. The Signal-to-Noise Ratio (SNR) assesses the intensity and quality of GPS signals received. Higher SNR levels are often associated with a more dependable and accurate signal. By using SNR as an input characteristic, the NN may detect inconsistencies or abnormalities that could indicate GPS spoofing efforts. Doppler shift refers to the change in frequency of the GPS signal induced by the UAV's relative motion to the GPS satellites. This data may be used to calculate the velocity of the UAV. The NN may identify discrepancies or anomalous data that may suggest GPS spoofing by using the doppler shift as an input feature.

The authors of the cited paper [22] conducted a comparative investigation of numerous machine learning (ML) algorithms for detecting jamming attacks in wireless networks. Their goal was to find the best effective method for detecting and neutralizing these disruptive signals. To do this, the scientists investigated several signal properties that might be suggestive of jamming activity. When wireless transmissions are jammed, these features capture certain traits or patterns. The ML models can learn to discriminate between normal network activity and the presence of jamming signals by examining these properties, and the Random Forest, SVM, and NN models were used. In [23] The authors suggest utilizing a multi-layer neural network (NN) to identify GPS signal faking. This method makes use of NNs' capacity to understand complicated patterns and correlations in data. While the authors recognize the usefulness of these strategies, they point out a restriction in terms of labeled data availability, particularly in

the case of attack data. Labeled data is essential for training supervised ML models such as the multi-layer NN described in the article. It is made up of samples with labels that indicate whether they are real GPS signals or faked signals. In reality, however, acquiring a substantial volume of labeled attack data might be difficult. This limitation prevents supervised ML approaches for detecting GPS signal spoofing from being widely used. It is difficult to build good and trustworthy algorithms that can discriminate between regular and faked GPS signals without a large annotated dataset.

To address the constraints of limited labeled data, researchers have resorted to unsupervised machine learning (ML) methods, notably anomaly detection algorithms. In the study [24], an autoencoder, a form of NN especially built for data reduction and reconstruction, is presented for defect detection in UAVs. The autoencoder model is trained to understand the regular patterns and traits of UAV behavior using a range of features. These features span across five classifications, including internal measurements, such as sensor readings, as well as external factors like location, position, orientation, system status, and control information. n [25], an unsupervised multivariate Gaussian-based anomaly detection system is applied to discover atypical driving behaviors in semi-autonomous cars. This algorithm focuses on analyzing data collected from accelerometer and GPS sensors in manually driven automobiles.

The algorithm is unsupervised, which means it does not rely on pre-labeled data indicating normal or abnormal behavior. Instead, the multivariate Gaussian distribution is used to model normal driving patterns based on sensor data. By capturing the statistical properties of the accelerometer and GPS readings during regular driving, the algorithm establishes a baseline for what is considered typical behavior. [26] employs an unsupervised multivariate Gaussian-based anomaly detection algorithm to detect unusual driving behaviors in semi-autonomous vehicles. This algorithm focuses on analyzing data collected from accelerometer and GPS sensors in manually driven automobiles. The algorithm is unsupervised, which means it does not rely on pre-labeled data indicating normal or abnormal behavior. Instead, the multivariate Gaussian distribution is used to model normal driving patterns based on sensor data. The algorithm establishes a baseline for what is considered typical behavior by capturing the statistical properties of the accelerometer and GPS readings during regular driving. The method compares real-time sensor data from semi-autonomous cars to the predefined typical behavior model during operation. Any major abnormalities from the revealed patterns are marked as odd driving behaviors. These abnormalities may signal possible safety issues, inappropriate vehicle operation, or unusual driving situations that necessitate additional examination or action.

Chapter 3 - Detecting and Countering Cyber-Attacks in CAVs

3.1 Introduction

Concerns about the susceptibility of CAVs to cyber-attacks are developing as their use grows. Because CAVs rely largely on digital networking and communication systems, they are appealing targets for hackers who can exploit software, hardware, or network flaws. Cyber-attacks against CAVs can include illegal access, manipulation of essential systems, data theft, or even remote control of the vehicle, highlighting the importance of strong security measures to safeguard passengers and maintain public safety. Attacks against the GPS receiver of CAVs, particularly spoofing attacks on GPS positions, are a major source of worry. The compromise of the GPS security of CAVs poses a serious risk to both drivers and pedestrians. The respective chapter comprises of the studies about how to detects such attacks in autonomous vehicles.

3.2 Methodology

There are numerous essential entities engaged in the situation of GPS location spoofing. The CAV and the attacker are the key players. There are also two infrastructure components: GPS satellite infrastructure and wireless network infrastructure. The CAV is a vehicle that uses GPS signals to find its location and move independently. It employs GPS receivers to receive signals from GPS satellites and determine its position based on the information received. The attacker intends to modify the GPS signals received by the CAV as part of the spoofing attack. The GPS satellite infrastructure is made up of a network of satellites that orbit the Earth and emit signals that GPS receivers use to derive accurate positional information. These satellites constantly broadcast signals providing time and location information. The wireless network infrastructure is the communication backbone that allows numerous devices, including the CAV and the attacker, to communicate with one another. Telecommunication firms' cellular towers, Wi-Fi access points, routers, and other networking equipment are all part of this infrastructure. It offers the wireless connectivity required for data transmission and communication between organizations. Figure 1 depicts the linkages and interactions between these entities in the GPS location spoofing scenario, with a graphical depiction of their responsibilities and connections.

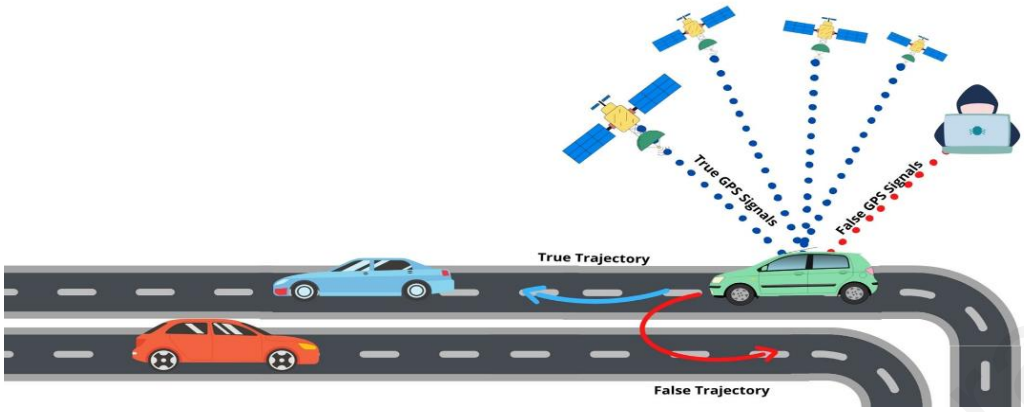


Figure 1. GPS Spoofing attack scenario on CAV

The statement's system model relates to a CAV traveling on a road network. The model considers two critical factors: the CAV's real position and velocity. The CAV's real position is represented by a vector $p_k = [x_k, y_k]^T$, where $x_k$and y_k are the CAV's coordinates at a single time instance marked by k. For mathematical simplicity, the "T" superscript indicates the transpose operation, which turns the row vector to a column vector. Similarly, the CAV's velocity is represented by a vector $u_k = [\dot{x}_k, \dot{y}_k]^T$, where $\dot{x}_k$ and $\dot{y}_k$ are the derivatives (rates of change) of the $x_k$and y_k coordinates. The dot above the variables indicates the time derivative, which shows how the position coordinates vary over time.

CAV is outfitted with a GPS receiver that receives satellite signals and delivers the vehicle's GPS location, abbreviated as $p_k^G = [x_k^G, y_k^G]^T$. In the context of prospective attacks, an attacker can utilize off-the-shelf equipment, such as SDR hardware, amplifiers, and antennas, as well as open-source SDR software, to interfere with and modify legal GPS signals. The attacker can carry this equipment on the ground or mount on an UAV. It is assumed in this study that the attacker spoofs the GPS position by providing a constant bias value in both GPS location coordinates. The GPS position of the attacked CAV is treated as Gaussian random variable, $p_k^G \sim \mathcal{N}[p_k + B_A, \sum_k^G]^T$. , where $\sum_k^G = diag_2(\sigma_k^G)$ is the covariance matrix of the GPS readings with equal standard deviation $\sigma_k^G = \sigma^G$ in both coordinates.

The bias provided by the attacker is represented by the attack vector $B_A = b[1,1]^T$, whereas b which indicates the size of the attack in meters for each point. A value of $b = 0$ implies that no attack is present. When the CAV is within range of the attack, the GPS receiver provides a spoofed position (i.e., $b \neq 0$), which the CAV interprets as its true location unless it has a reliable method to detect and counteract spoofing attempts. In other words, if suitable detection mechanisms are not implemented, the CAV will accept the spoof location as real during the attack.

The CAV is outfitted with a specific technology that can estimate position independently of GPS. This gadget monitors signals from the surrounding wireless network infrastructure and other linked cars using SDR hardware and software. It employs a Localization Algorithm (LA) to predict the CAV's present location based on these signals. The LA can use network-assisted methodologies, such as measuring time, angle, or signal intensity from nearby transmitters like as cellular towers or Wi-Fi access points. Cooperative LAs take use of vehicle collaboration to increase location estimate accuracy. This second device improves the CAV's capacity to detect its location, providing redundancy and possibly acting as a backup in instances when GPS signals are inaccurate or corrupted.

The CAV's specialized device offers an estimated position for the vehicle, which is indicated as $p_k^L = [x_k^L, y_k^L]^T$. This estimated position is represented by a Gaussian random variable, $p_k^L \sim \mathcal{N}[p_k, \sum_k^L]^T$, where pk is the real location of the CAV and $\sum_k^L = diag_2(\sigma_k^L)$ is the covariance matrix that describes the uncertainty or noise associated with the estimation. The standard deviation of the noise in the Localization Algorithm (LA) calculated CAV positions is represented as $\sigma_k^L = \sigma^L$ in both coordinates. The variability and uncertainty in the predicted locations supplied by the LA are captured by this Gaussian modeling. It admits that estimated positions may differ from genuine locations owing to a variety of causes like as measurement mistakes, signal interference, or algorithmic flaws. The magnitude of this uncertainty is reflected in the covariance matrix, with bigger values indicating greater uncertainty in the calculated locations. The system may accommodate for the inherent constraints and uncertainty in the localization process by including this probabilistic representation, resulting in more robust and dependable location-awareness for the CAV.

3.3 Attack Detection Framework

3.3.1 Overview

Figure 2 depicts the suggested method for detecting GPS spoofing attacks, which includes an in-vehicle assessment of GPS location integrity. During the Prediction phase, the onboard sensor readings collected through the OBU or CAN bus are used to anticipate the CAV's location at time k+1. This forecast, designated as $\hat{p}_{k+1} = [\hat{x}_{k+1}, \hat{y}_{k+1}]^T$, is based on the CAV's prior revised location estimate, $\tilde{p}_k = [\tilde{x}_k, \tilde{y}_k]^T$. The position is projected ahead of time by combining sensor readings and utilizing the CAV's underlying mobility model. The CAV's GPS-free position measurements, referred to as p_{k+1}^L, derived through the LA, are utilized to refine the anticipated location estimate during the Update phase. This refinement is accomplished using Bayesian filtering techniques, which utilize the fresh measurements to

update the anticipated position and produce a more precise estimate of the CAV's location at time $k+1$, indicated as $\tilde{p}_{k+1}$.

The suggested approach improves the accuracy and reliability of the CAV's position estimation by integrating the Prediction and Update stages. It refines the anticipated position estimate using sensor readings and GPS-free location measurements, thereby reducing the effects of GPS spoofing attacks. This method allows the CAV to keep a consistent knowledge of its position, which is critical for safe and effective autonomous driving operations.

The GPS position measurements collected from the vehicle's GPS receiver, p_{k+1}^{G}, are compared to the refined location estimate, $\tilde{p}_{k+1}$, during the Attack Detection phase. If the difference between them exceeds a predetermined threshold T_d, an alert is triggered, indicating the detection of a GPS spoofing attempt. The algorithm below describes the complete procedure. If an attack is discovered, one mitigating method is for the CAV to stop utilizing GPS data and instead depend on the revised position estimate, $\tilde{p}_{k+1}$, for location-based functions like as navigation. This guarantees that the CAV's actions are not based on compromised GPS data and that it can continue to execute dependable and precise operations even in the midst of an attack.

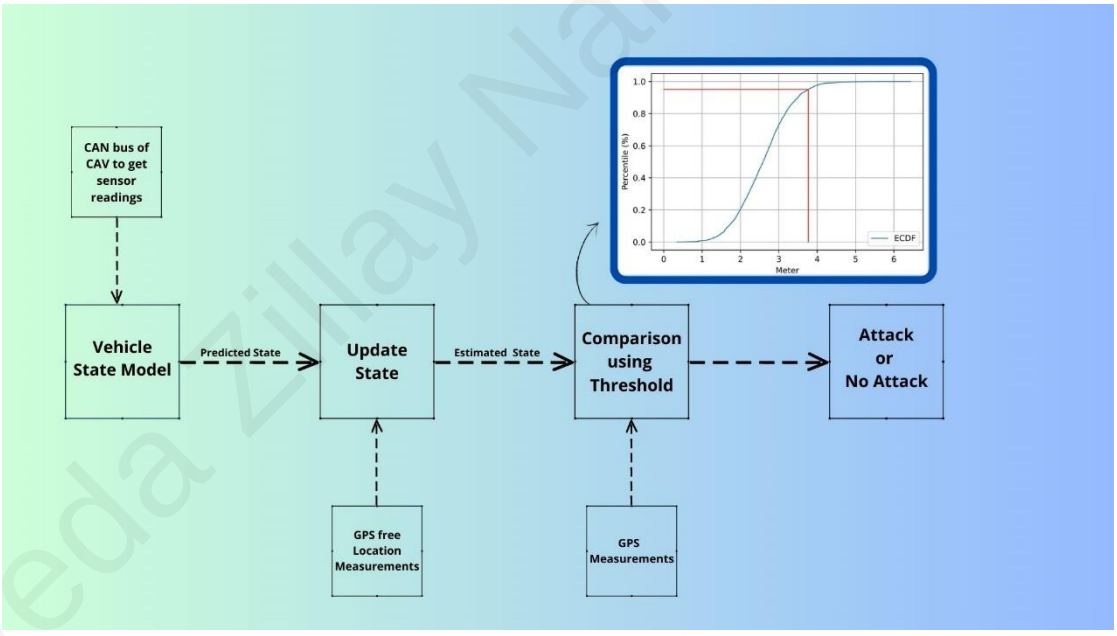


Figure 2. In-Vehicle attack detection data flow framework

Below is the algorithm developed to detect GPS spoofing attack.

Input: *previous location estimates* $[\tilde{p}_k, \sum_k^{\tilde{p}}]$, *CAV's sensory data* $(\alpha, \varphi, \dot{v})$,

radio signal data, GPS location $[p_{k+1}^{G}, \sum_{k+1}^{G}]$, *windowsize* ω *and threshold* T_d

Output: *GPS Spoofing Attack Detection*

1. $[\hat{p}_{k+1}, \sum_{k+1}^{p}]$ ←EKF predicted $(\tilde{p}_k, \alpha, \dot{\varphi}, v)$
2. $[\mathrm{p}_{k+1}^{L}, \sum_{k+1}^{L}]$ ←LA (radio signal data)
3. $[\hat{p}_{k+1}, \sum_{k+1}^{p}]$ ←EKF update($[\hat{p}_{k+1}, \sum_{k+1}^{p}], [\mathrm{p}_{k+1}^{L}, \sum_{k+1}^{L}]$)
4. d_{k+1}^{E} ←distance ($[\hat{p}_{k+1}, \sum_{k+1}^{p}], [\mathrm{p}_{k+1}^{L}, \sum_{k+1}^{L}]$)
5. d_{k+1}^{E} ←filter $(d_{k-\omega+2}^{E}, \ldots., d_{k+1}^{E})$
6. $if\ d_{k+1}^{E} > \mathrm{T}_{d}^{E}$ then GPS location spoofing attack detected.

3.3.2 Prediction Phase

During the prediction phase, our approach makes use of data from the vehicle's onboard sensors, such as steering angle (α), yaw rate ($\dot{\varphi}$), and wheel speed (v). Using these sensor inputs, our approach forecasts the vehicle's future position, indicated as $\hat{\mathrm{p}}_{k+1}$, inside a time step of Δt. This prediction is based on the use of the well-known bicycle model [39], which is a nonlinear model that represents the vehicle's system state using fundamental physics rules. The one-step forecast of the vehicle's position and speed in its body-frame reference system may be calculated by assuming the vehicle's body-frame aligned with the x-axis.

$$\begin{pmatrix} x_{k+1}^{u} \\ \dot{x}_{k+1}^{u} \\ y_{k+1}^{u} \\ \dot{y}_{k+1}^{u} \end{pmatrix} = \begin{pmatrix} \upsilon\Delta\mathrm{t} \\ \upsilon \\ \frac{1}{2}(C_f\left(\alpha - \frac{l_{f\dot{\varphi}}}{\upsilon}\right) + C_r\frac{l_{r\dot{\varphi}}}{\upsilon})\frac{1}{M}\Delta\mathrm{t}^2 \\ (C_f\left(\alpha - \frac{l_{f\dot{\varphi}}}{\upsilon}\right) + C_r\frac{l_{r\dot{\varphi}}}{\upsilon})\frac{1}{M}\Delta\mathrm{t} \end{pmatrix} \ldots\ldots\ldots\ldots (1)$$

In equation (1) the variables x_{k+1}^{u} and y_{k+1}^{u} in the current context reflect the longitudinal and lateral displacements (velocities) between two successive time steps in the vehicle's body frame. The parameters l_f and l_r represent the front and rear wheel distances from the vehicle's barycenter, respectively. M denotes the vehicle's mass, whereas C_f and C_r denote the corner stiffness of the front and rear wheels, respectively.

We may acquire a one-step prediction of the vehicle's position in the global geographic reference system after conducting a coordinate transformation. This transformation enables us to change the expected position from the body frame of the vehicle to the global frame:

$$\begin{pmatrix} \hat{x}_{k+1} \\ \hat{\dot{x}}_{k+1} \\ \hat{y}_{k+1} \\ \hat{\dot{y}}_{k+1} \\ \hat{\varphi}_{k+1} \end{pmatrix} = \begin{pmatrix} \tilde{\mathrm{x}}_k + x_{k+1}^{u}cos\hat{\varphi}_k - y_{k+1}^{u}sin\hat{\varphi}_k \\ \dot{x}_{k+1}^{u}cos\hat{\varphi}_k - \dot{y}_{k+1}^{u}sin\hat{\varphi}_k \\ \tilde{y}_k + x_{k+1}^{u}cos\hat{\varphi}_k - y_{k+1}^{u}sin\hat{\varphi}_k \\ \dot{x}_{k+1}^{u}sin\hat{\varphi}_k - \dot{y}_{k+1}^{u}cos\hat{\varphi}_k \\ \hat{\varphi}_k + \dot{\varphi}\Delta\mathrm{t} \end{pmatrix} \ldots\ldots\ldots\ldots (2)$$

We employ the Extended Kalman Filter (EKF) technique to estimate the covariance of the vehicle's system state, assuming that the measurement noise is uncorrelated and has a Gaussian distribution. This is accomplished by applying the EKF method to equations (1) and (2) in [39].

The predicted position, $\hat{p}_{k+1}$, is represented as $\hat{p}_{k+1} \sim \mathcal{N}[\hat{p}_{k+1}, \sum_{k+1}^{p}]$, where $\sum_{k+1}^{p}$ represents the covariance matrix that quantifies the uncertainty associated with the anticipated location.

3.3.3 Update Phase

The EKF method combines the anticipated vehicle's position, $\hat{p}_{k+1}$, with the GPS-free global location measurement, p_{k+1}^{L}, received from the LA utilizing radio signal data during the update phase. As seen in Algorithm in lines 2-3, this fusion process yields a revised location estimate, $\tilde{p}_{k+1}$. The improved position estimate has a Gaussian distribution, which is represented by $\tilde{p}_{k+1} \sim \mathcal{N}[\tilde{p}_{k+1}, \sum_{k+1}^{\tilde{p}}]$,, where $\sum_{k+1}^{\tilde{p}}$ is the covariance matrix representing the uncertainty associated with the refined location estimate.

3.3.4 Attack Detection Phase

The basic idea behind detecting GPS location spoofing attempts is to use a distance metric to compare the estimated position of the CAV, $\tilde{p}_{k+1}$, with the GPS reading, p_{k+1}^{G}, and see if the divergence exceeds a predetermined threshold. This approach is carried out in four steps, the first of which is undertaken offline in attack-free settings to identify a suitable threshold value. Steps 1-3 are carried out live while the attack detection solution is running within a moving CAV under unknown conditions, as shown in Algorithm.

The detection technique entails continually analyzing the difference between the estimated and GPS positions, and if this difference exceeds a certain threshold, an alert is raised to signal the presence of a GPS location spoofing assault. Step 0's threshold selection achieves a balance between detecting legitimate threats and reducing false alarms. Steps 1-3 are carried out in real-time while the CAV is in motion, allowing for rapid identification and reaction to possible threats.

Threshold Selection Phase:

A data-driven technique is used to establish the threshold value T_d. In the beginning, an attack-free time is considered, during which a CAV equipped with the proposed solution collects a series of N GPS position measurements, p_n^G, and corresponding estimated locations, $\tilde{p}_n$, using the EKF method. In Step 1, the pairwise distances between each location pair, d_n, are calculated using two candidate distance metrics. In Step 2, a filtering technique is used to eliminate noise in the distance measurements, resulting in the filtered distance values, $\bar{d}_n$. Finally, based on the filtered distances, $\bar{d}_n$, the Empirical Cumulative Distribution Function (ECDF) is created. The threshold value T_d is determined by calculating the ECDF curve's γ^{th} percentile using the parameter $\gamma \in [0\ 1]$ which spans from 0 to 1. Based on the distribution of distance values, this empirical technique enables the selection of a suitable threshold.

The parameter selection introduces a trade-off in the detection system's performance, especially in terms of proper detection and false alarm rates when attacks are present. A low number corresponds to a low T_d, which increases the chance of detecting possible attacks. However, this increases the number of false alarms, making the method less effective. Choosing a high number, such as $\gamma = 1$, on the other hand, assists in reducing the issue of false alarms, but at the expense of potentially missing certain attack detections, increasing the rate of missed detections. As a result, selecting entails balancing the trade-off between properly identifying threats and limiting false alarms.

Location Distance Computation:

In this work, we investigate distance metrics, namely the Euclidean distance d_{k+1}^{E}, to calculate the difference between the predicted CAV position $[\tilde{p}_{k+1}, \sum_{k+1}^{\tilde{p}}]$, and the GPS location $[p_{k+1}^{G}, \sum_{k+1}^{G}]$. It is possible to use it, as mentioned in line 4 of Algorithm, although alternative distance metrics can also be used. We have removed the time index k to simplify the notation. The straight-line distance between two places is computed using the Euclidean distance. Euclidian Distance is given as:

$$d^{E}(\tilde{p}, p^{G}) = \|\tilde{p} - p^{G}\|$$

Location Distance Filtration:

Due to inherent noise impacting both the GPS and estimated locations, the distance value d_{k+1}^{E} obtained in Step 1 might fluctuate dramatically throughout the CAV's travel, even for neighboring places. Environmental variables (for example, cloud cover and humidity) add to noise in GPS data. GPS noise may be stronger in urban locations with tall structures than in suburban or rural regions with better satellite view. Furthermore, because of the restricted radio signals available for localization in places with scant infrastructure, the estimated positions p_{k+1}^{L} from the LA may have greater noise. As a result, these uncertainties permeate the estimated distance value, raising the possibility of erroneous attack detection.

Using covariance matrices $\sum^{\tilde{p}}$ and $\sum_{k+1}^{G}$, the uncertainty in d_{k+1}^{E} may be evaluated and quantified. However, it is critical to reduce noise in d_{k+1}^{E} for a more robust detection method. A filtering approach is used to do this, as demonstrated in line 5 of Algorithm. A sliding window averaging filter α is employed specifically, with a window of size ω processing the previous distance data to compute the filtered distance. This filtering helps to smooth out noise fluctuations and improves the accuracy of the distance measurement. Filtered distance is calculated as below:

$$d_{k+1}^{E} = \frac{\sum_{i=k-\omega+2}^{k+1} d_{i}^{E}}{\omega}$$

Decision for Detection (Attack/No Attack):

The last step in identifying a GPS location spoofing attempt is to compare the filtered distance value, $\bar{d}^{E}_{k+1}$, acquired in Step 2 with the threshold value, T^{e}_{d}, determined in Step 0. If the filtered distance value exceeds the threshold, it indicates that a GPS location spoofing attack has been detected. Lines 6-7 of Algorithm depict this choice.

3.4 Sensor Calibration and testing

In this section we will discuss calibration of three sensors that we used for the process of prediction and estimation of location of the CAV. The Figure 3 shows the CAV and different sensors.

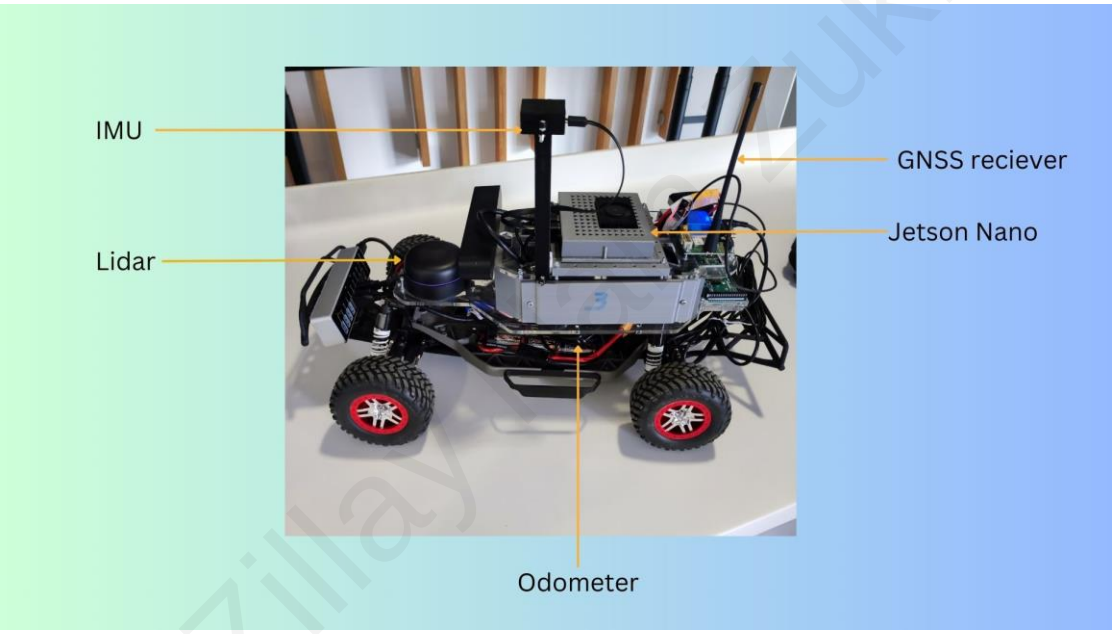


Figure 3. CAV at KIOS Lab mounted with different sensors

3.4.1 IMU Sensor Calibration:

IMU measures and tracks an object's orientation, velocity, and acceleration in three dimensions. It is normally made up of three major components: an accelerometer, a gyroscope, and, in certain cases, a magnetometer. The accelerometer detects changes in velocity and location by measuring linear acceleration. The gyroscope monitors angular velocity and provides information about the rotation of an item. When present, the magnetometer monitors the intensity and direction of the magnetic field to help determine the object's orientation relative to the Earth's magnetic field. IMU sensors are widely employed in a wide range of applications, including robots, virtual reality, navigation systems, and motion tracking. To calibrate this sensor please check Appendix A. The output after testing IMU are shown in Figure 4 below.

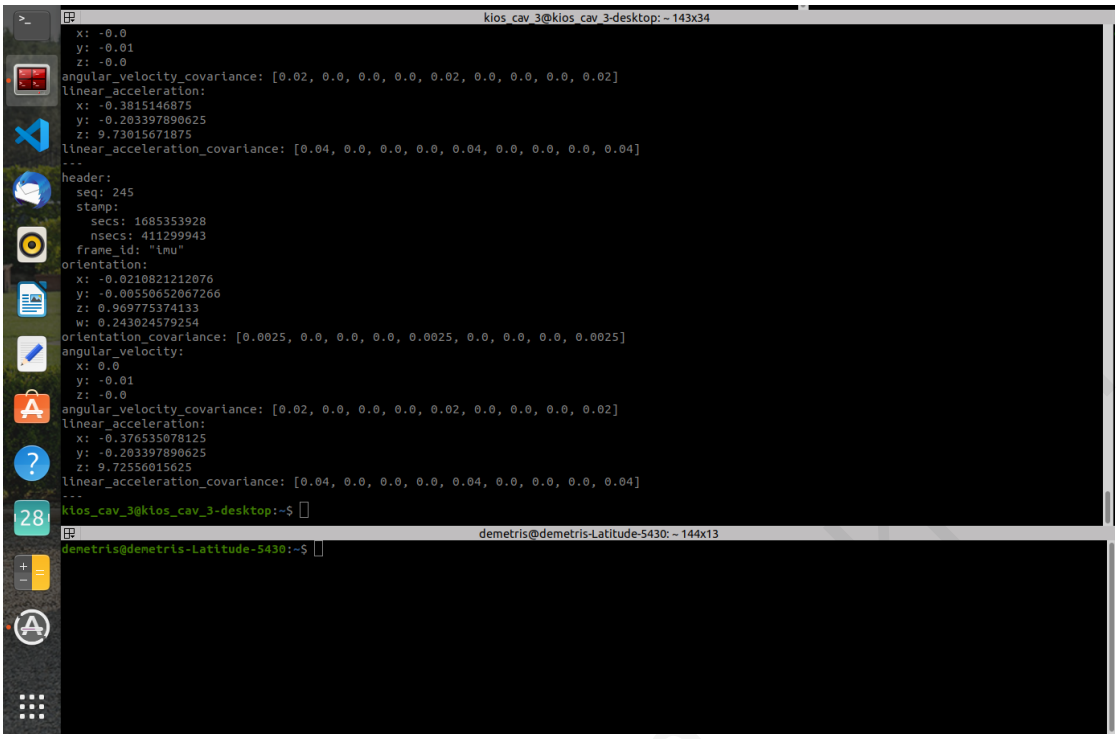


Figure 4. IMU sensor configured results.

3.4.2 VESC Sensor Calibration:

An odometry sensor, also known as wheel encoders or wheel odometry, is a sensor that measures the rotation of a robot's or vehicle's wheels to determine its velocity. It gives information regarding the vehicle's distance traveled, direction, and velocity. The sensor is normally made up of one or more encoders that are installed on the vehicle's wheels. The encoders emit pulses when the wheels revolve, which are counted and utilized to compute the vehicle's movement. The odometry sensor can assess the location and orientation of the vehicle relative to its starting point by integrating these readings over time. Odometry sensors are widely utilized in robots, self-driving cars, and navigation systems.

Configuration of the VESC motor:

If you have problems with the motor, such as "crack noises at low speeds," you may need to configure the PID (Proportional-Integral-Derivative) parameters. This may be accomplished by changing the PID parameters in the VESC Tool to maximize motor performance.

To check if the calibrated sensor is working find a couple of tests were conducted. The CAV was set to move in a circular trajectory for some time and data from the VESC was gathered like time, speed, steering angle, x (in meters) and y (in meters). The center of circle was not defined at (0,0) so we reconstructed the path and by calculating the path of the circle theoretically we concluded. The theocratically constructed path was same as the one constructed

based on the original values from the VESC as shown in Figure 5 and results are shown in Figure 6.

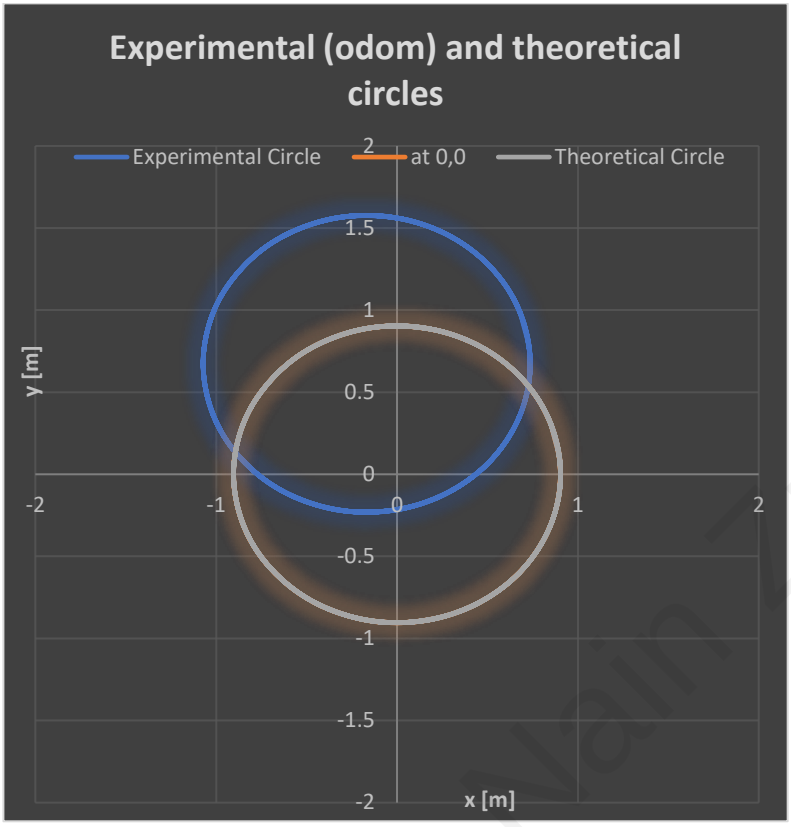


Figure 5. Experimental Odometry and Theoretical Trajectory comparison.

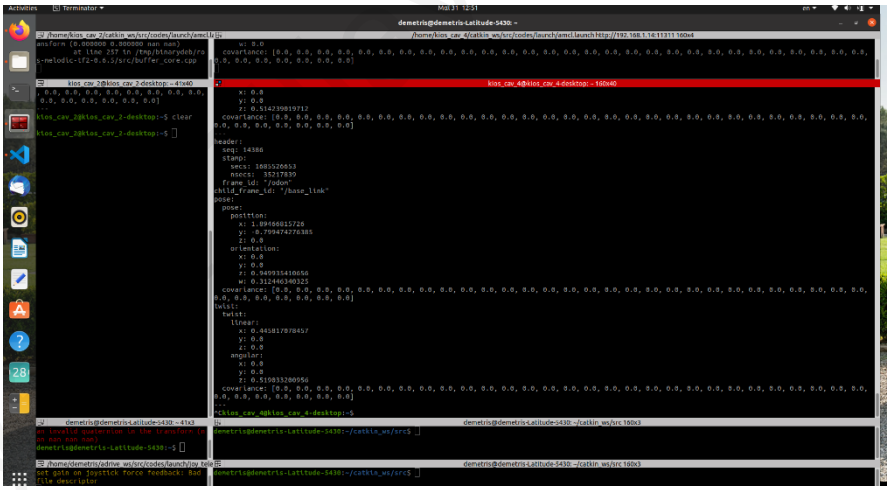


Figure 6. VESC sensor test results.

3.4.3 GNSS sensor calibration

A GNSS sensor is a device that collects signals from numerous satellite constellations in order to calculate precise location, navigation, and time. It provides position data for many applications like as navigation systems, mapping, and surveying by using signals from systems such as GPS, GLONASS, Galileo, and BeiDou. The u-blox C94-M8P is a high-precision GNSS

sensor used for accurate location on CAVs (Connected Autonomous Vehicles). It uses RTK technology, which allows for centimeter-level positional precision.

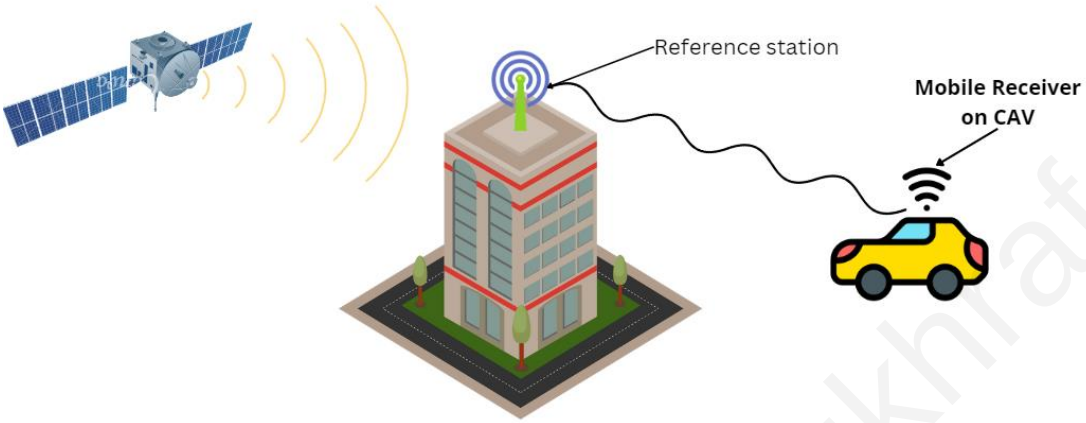


Figure 7. GNSS calibration process.

The calibration process include calibration the reference station of GNSS is placed in the open air and calibrated there as it is supposed to provide the exact location as shown in Figure 7. It is a reference for Mobile receiver mounted on the CAV. We calibrated the sensor based on the documentation on the link provided in reference [42].

3.5 Attack Generation

A thorough analysis was carried out in the thesis research to investigate the vulnerabilities of GPS systems to jamming and spoofing attacks. To replicate those attacks, RF signals were generated and sent using a HackRF device as shown in Figure 8, an SDR platform. The gadget was used to create deliberate interference signals in order to impair GPS signals, resulting in jamming effects. Spoofed GPS signals were also made to mimic the behavior of authentic GPS satellites. The goal of these attacks was to test GPS receiver durability and efficacy, as well as the effects of such attacks on navigation systems.

Figure 8. HackRF device used to generate the GPS Spoofing attack

It is critical to collect the necessary data, especially GPS readings, in order to produce GPS attacks. This is possible by obtaining Earth data from credible sources such as the NASA website [43]. The "GNSS Status" app may be used to ensure the availability of a suitable number of satellites for GPS data collecting. Using Figure 9, it is feasible to evaluate whether a sufficient number of satellites are in range to acquire GPS data. Registration on the respective website is required to obtain the most recent dataset.

These early efforts are critical because they provide the groundwork for next rounds of study to generate accurate and reliable GPS attacks. The NASA website data provides a credible and comprehensive source of Earth data, allowing the construction of realistic attack scenarios. The possibility of gathering GPS data is determined by validating the number of satellites using the "GNSS Status" app. By registering on the website, you have access to the most recent data set, which is essential for performing complete and up-to-date research on GPS attack generation.

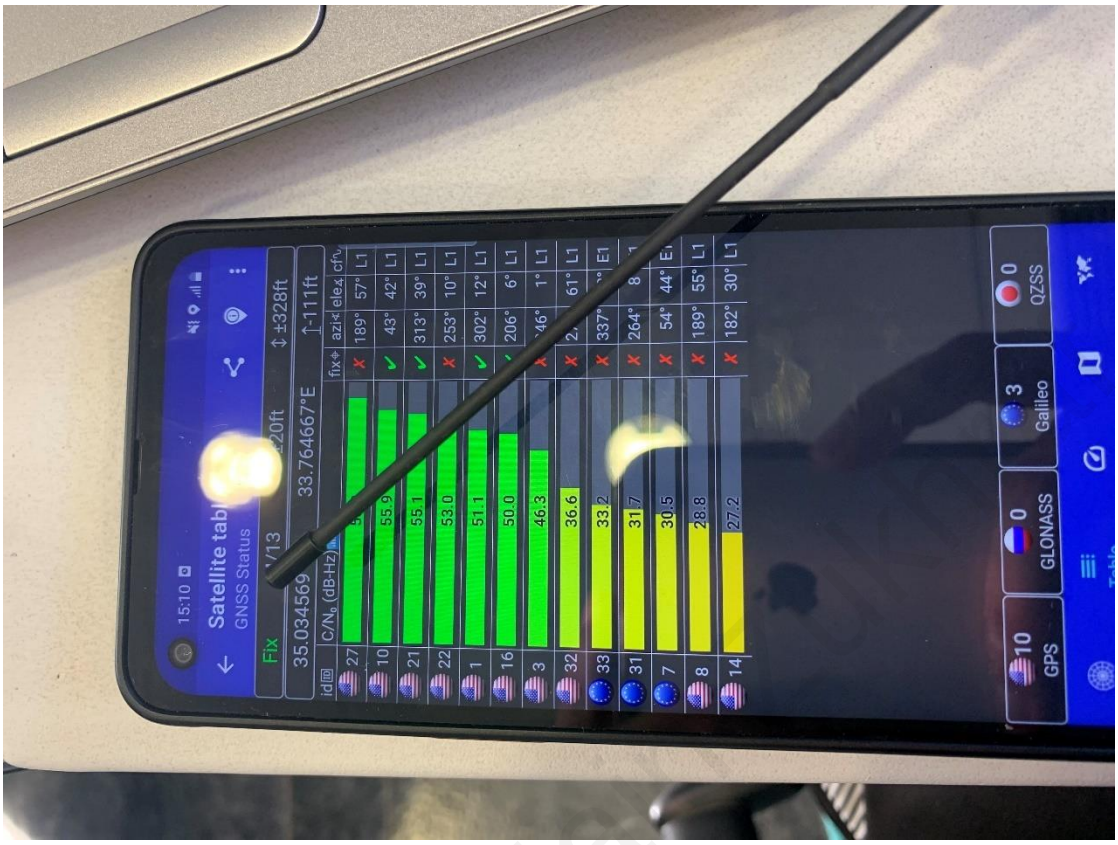


Figure 9. Connected Satellites are shown by using GNSS Status app.

After the initial stages were completed, a jamming attack was created to interrupt incoming GPS signals from satellites during the first 100 seconds. Following that, a GPS spoofing attack was conducted, as seen in Figures 10 and 11.

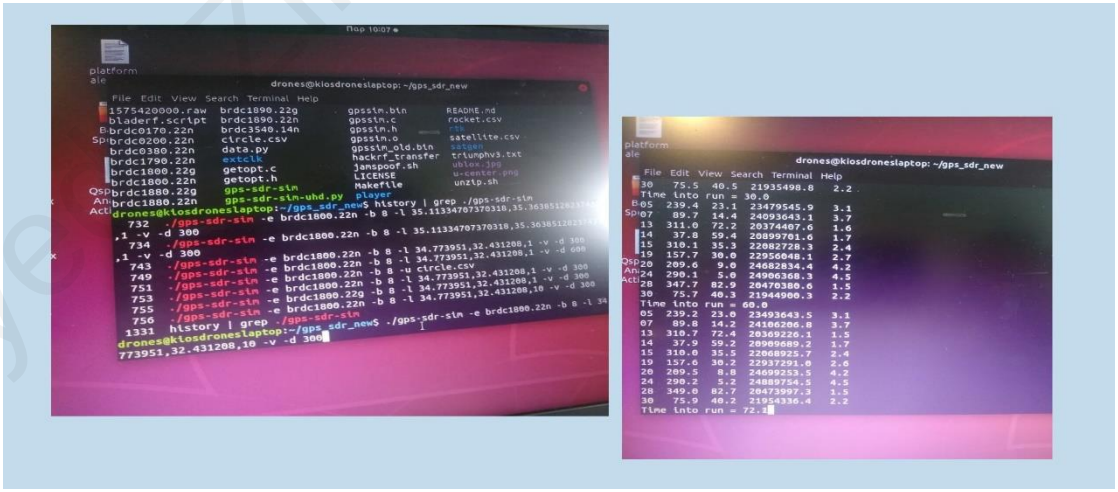


Figure 10. GPS Jamming attack launched

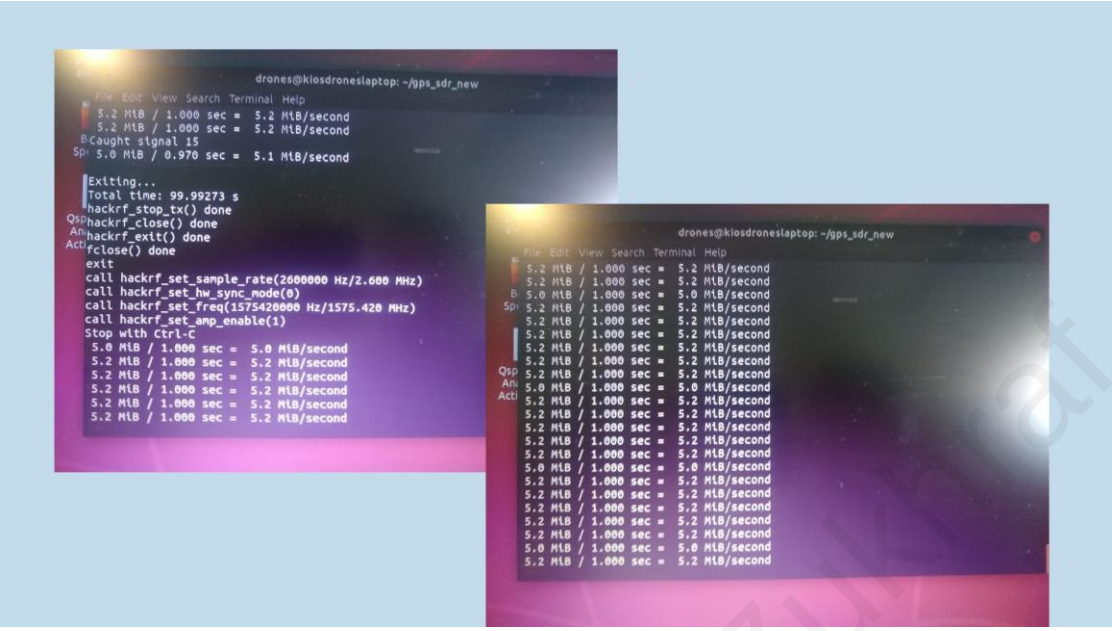


Figure 11. GPS Spoofing attack launched after Jamming attack.

The attack was generated on the remote controller of a drone at first. The remote was placed in the KIOS lab at the time of attack generation but the coordinates shown by the remote after attack was generated lied in the Makario Stadium in Nicosia Cyprus as shown in Figure 12.

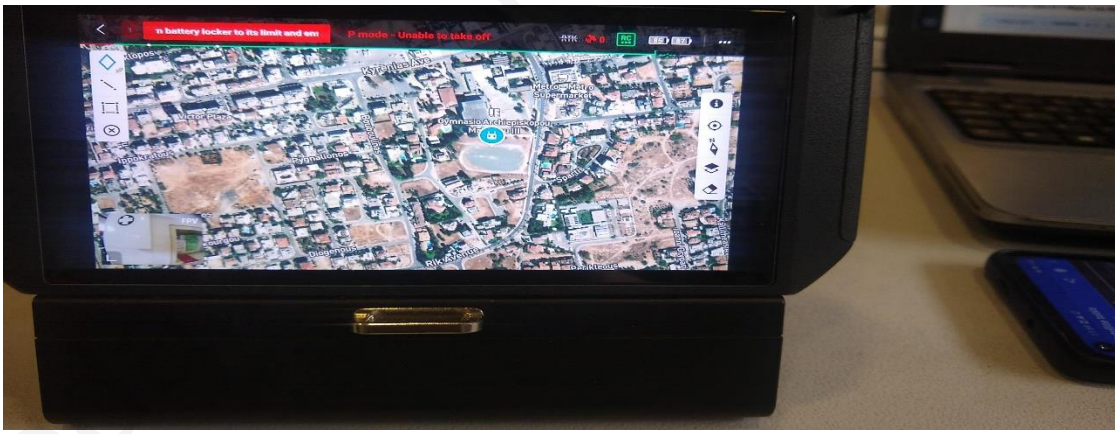


Figure 12. Location of the remote after attack generation.

3.6 Performance Validation

We used the CARLA simulator version 0.9.8, which is built on the Unreal Engine 4 platform, to evaluate the system's performance. CARLA is a powerful tool for developing, training, and validating autonomous driving systems. The simulations were done on a workstation PC running Linux (Ubuntu 18.04 Bionic) with 8 GB of RAM and a Vulkan-capable GPU. In addition, a Python 3.8 environment was used. All the data was published and Subscribed by using ROS environment (ROS Melodic 1.14.12), This setup guaranteed that the simulations

went smoothly and offered an ideal setting for achieving exact and dependable performance results throughout the system evaluation. The flow of information using ROS is illustrated in Figure 13.

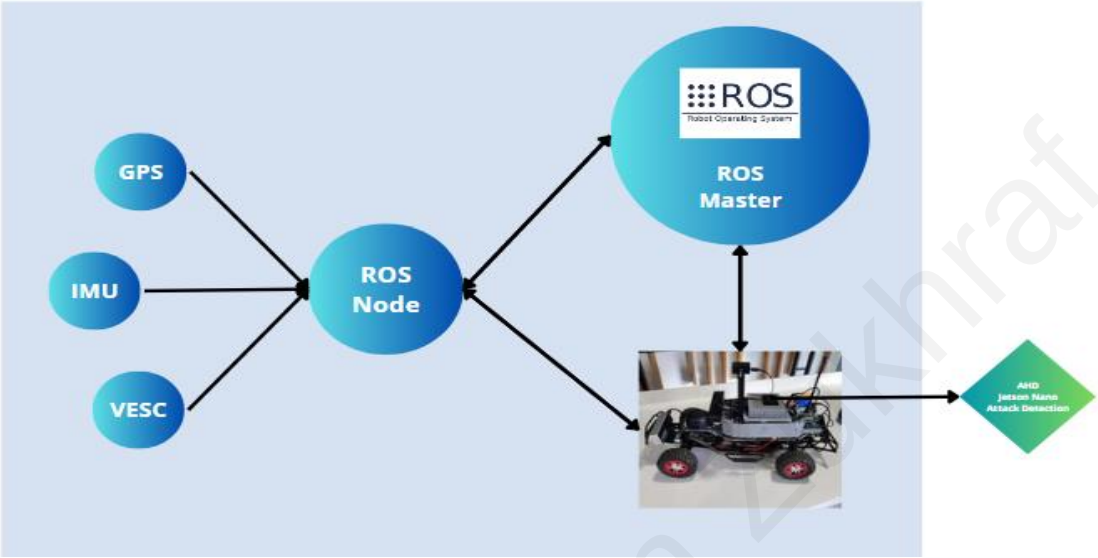


Figure 13. ROS framework data flow diagram.

3.6.1 Performance metrics

A confusion matrix, also known as a contingency table, is used to evaluate the detection results of the suggested solution. The outcomes are classified as True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN) in this matrix. The performance of the attack detection method is evaluated using multiple metrics based on these areas F1 Score and G mean. Precision (P) precisely gauges the algorithm's accuracy in identifying attacks. It reflects the percentage of real attacks identified to the overall number of detected attacks, including false positives, in our application scenario given by:

$$P = \frac{\text{number of true attack detected}}{nuber\ of\ true\ and\ false\ attack\ detected} = \frac{\mathrm{T}_p}{\mathrm{T}_p + \mathrm{F}_p}$$

Similarly, Recall (R) is the ratio of true attacks identified to all genuine attacks, including missed detections (false negatives).

$$R = \frac{\text{number of true attack detected}}{nuber\ of\ true\ \ attack\ detected} = \frac{\mathrm{T}_p}{\mathrm{T}_p + \mathrm{F}_N}$$

Finally, the F1 score is the weighted average of P and R and is used to assess data set accuracy.

$$F1 = 2\left(\frac{\mathrm{P} \times \mathrm{R}}{P + R}\right)$$

3.6.2 *Process pipeline*

Figure 14 depicts the simulation method, including the processes involved. Initially, the vehicle operates normally to set a threshold. The attack is then created using the HackRF device, and the attack detection system, which has previously been calibrated using the threshold calculation, detects it. Figure 15 depicts the approach that would be followed in real-world settings. However, due to GNSS sensor limitations and the high error rate of GPS measurements, the entire procedure could not be completed.

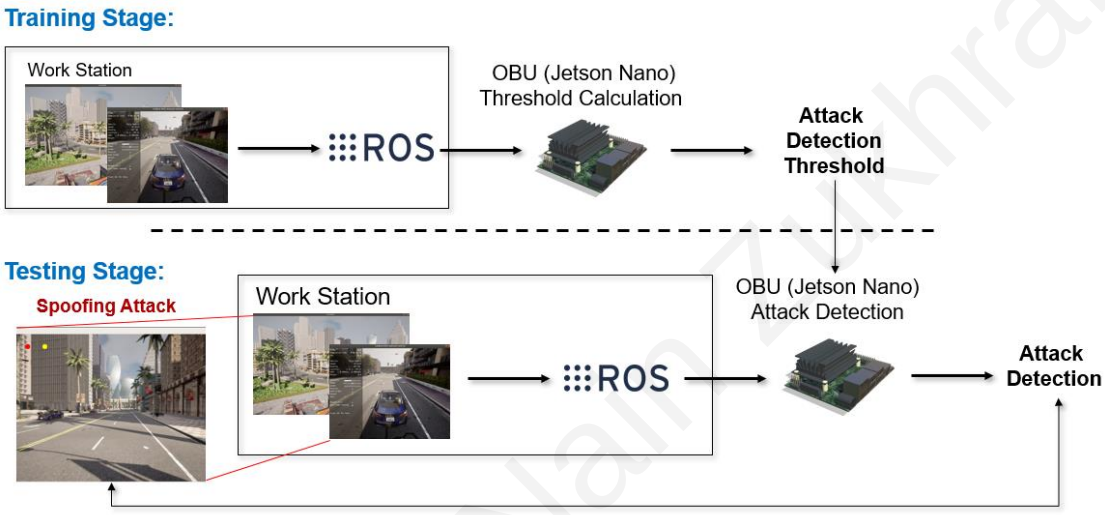


Figure 14. Pipeline for attack detection.

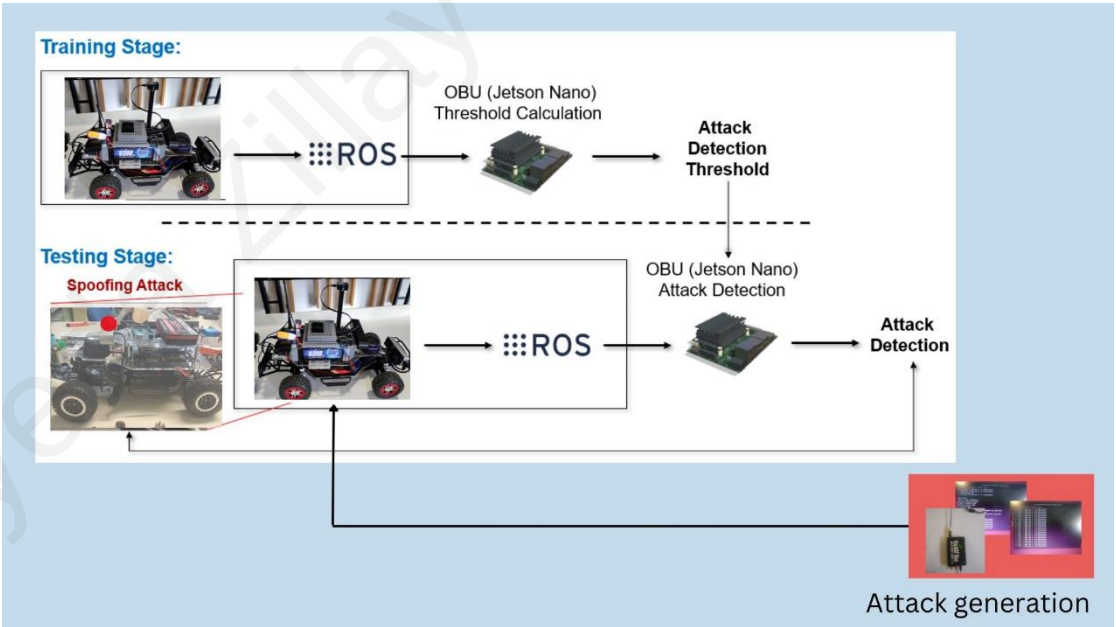


Figure 15. Real-world settings.

3.7 Experimental results

3.7.1 Effect of using sliding window generated by one dataset on other

In this section we performed a test to check how the attack detection algorithm works if we use one threshold form one data set to on the other data set and vice versa. The results are shown in the Figures 16, 17, 18 and 19.Specification of data set D_1 used:

length of data = 13479

Threshold Value Euclidian Distance Not Applying Sliding Window:4.6008

Threshold Value Euclidian Distance After Applying Sliding Window:3.5537

Specification of data set D_2 used:

Length of data = 10311

Threshold Value Euclidian Distance Not Applying Sliding Window:4.6617

Threshold Value Euclidian Distance After Applying Sliding Window:3.7838

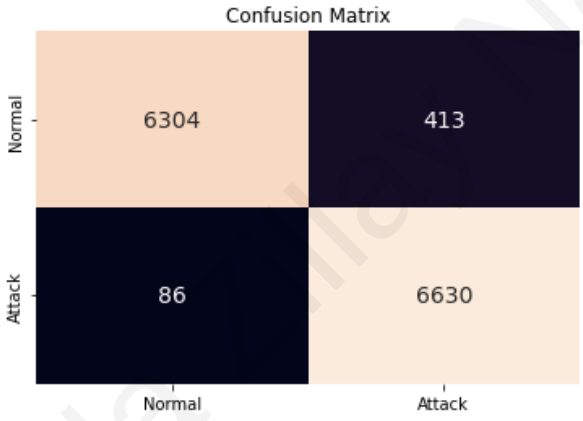


Figure 16. Result-I by using Dataset D_1 and threshold D_1^{th}=3.55

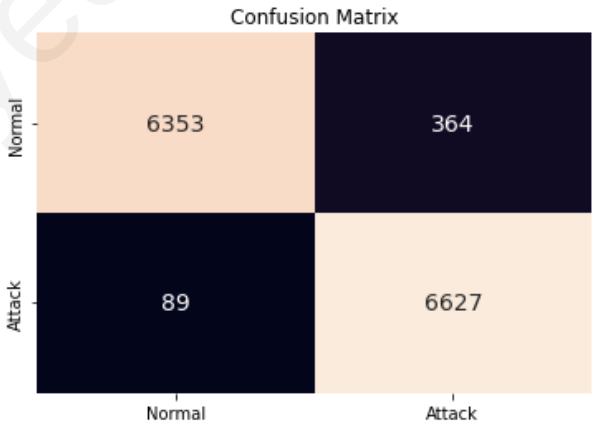


Figure 17. Result-I by using threshold D_1^{th} on D_2

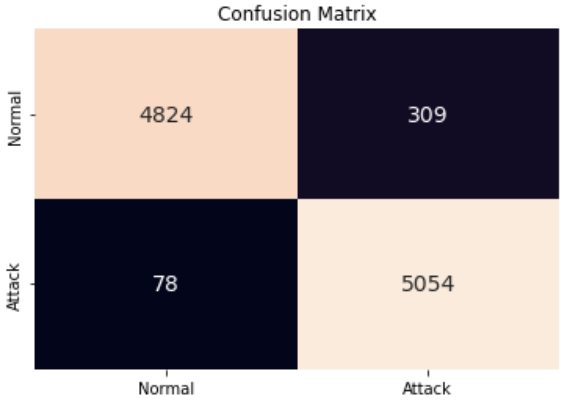


Figure 18. Result-I by using Dataset D_2 and threshold D_2^{th}=3.78

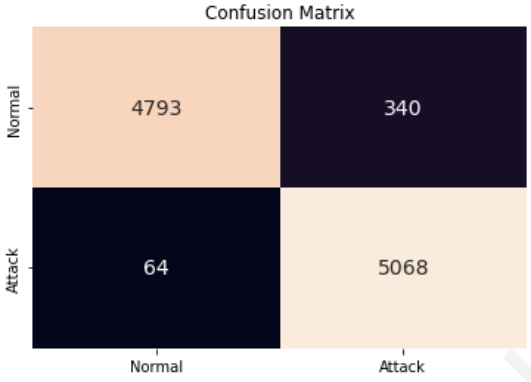


Figure 19. Result-I by using threshold D_2^{th} on D_1

By doing this analysis not a big difference is observed as the two-threshold obtained are not of big difference. Figure 16 shows results obtained by using dataset D_1 and threshold D_1^{th}=3.55. We tested the threshold obtained from trajectory to another. Figure 17 shows the results obtained by testing the above scenario. Like wise Figure 18 and Figure 19 depicts the scenario from second trajectory tested in the same manner.

3.7.2 Effect of different sliding window on the same dataset

Using a trajectory from a Carla simulator and gathering sensor data following analysis is being conducted the Figure 20(a) and (b) below shows the path followed by the CAV in simulator as Result-I for sliding window $\omega = 3$.

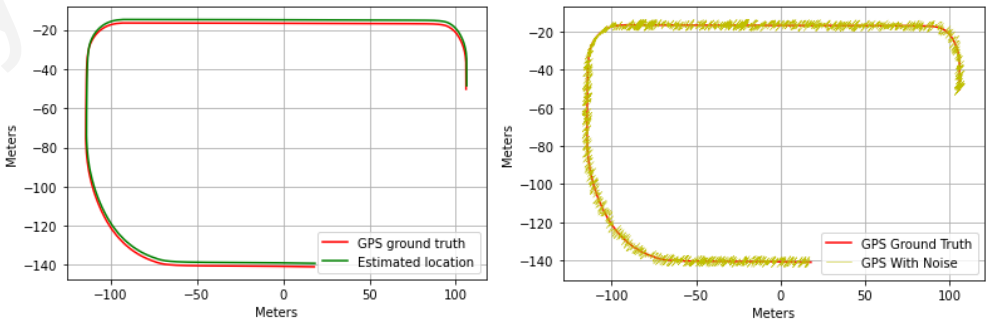


Figure 20. Result-I by using $\omega = 3$ (a) GPS ground Truth and estimated location (b) GPS Ground Truth with noise

By conducting analysis using confusion matrix as show in in Figure 21, Figure 22 and Figure 23 confusion matrix classifies the results into True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN).

Figure 21. Confusing matrix for Result-II

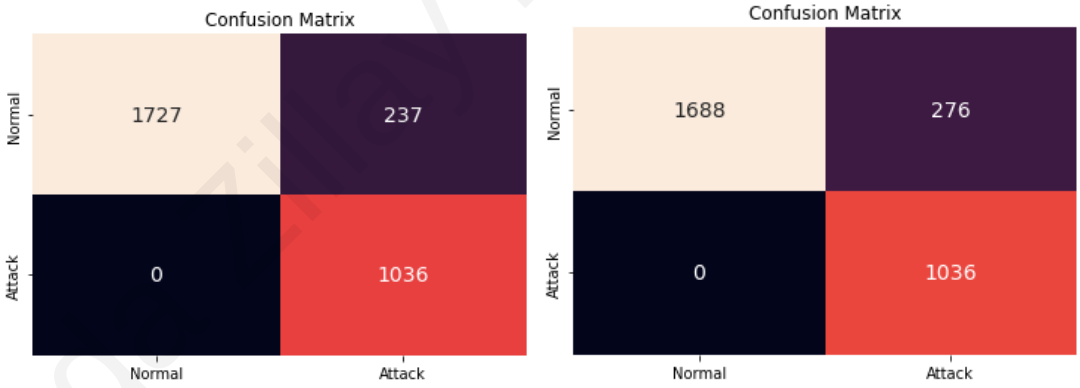


Figure 22. Result- II by using $\omega = 5$ (a) and $\omega = 10$ (b)

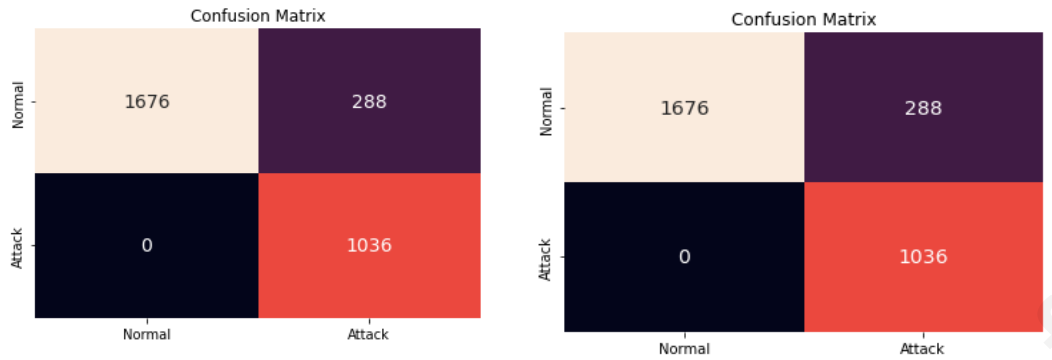


Figure 23. Result-II by using $\omega = 15$ (a) and $\omega = 20$ (b)

3.7.3 Testing Solution on Different Trajectories

In this section, different maps on CARLA Simulator were used to test the detection algorithm.

Trajectory 1:

Different parameter assigned before conducting the experiment are mentioned in TABLE 1.

Table 1. Parameter assignment for Map 1

	Parameter	**Value**
1	Length of data	11776
2	Attack bias	9
3	Sliding window	5
4	Standard Deviation of Signals of Opportunity	10
5	Standard Deviation of Signals of Opportunity Orientation	20
6	Standard Deviation of the Noise in the GPS	3
7	Sampling Interval in seconds	0.05
8	Percentile for threshold	95.0
9	Town name (From Carla)	Town 10

Graphs:

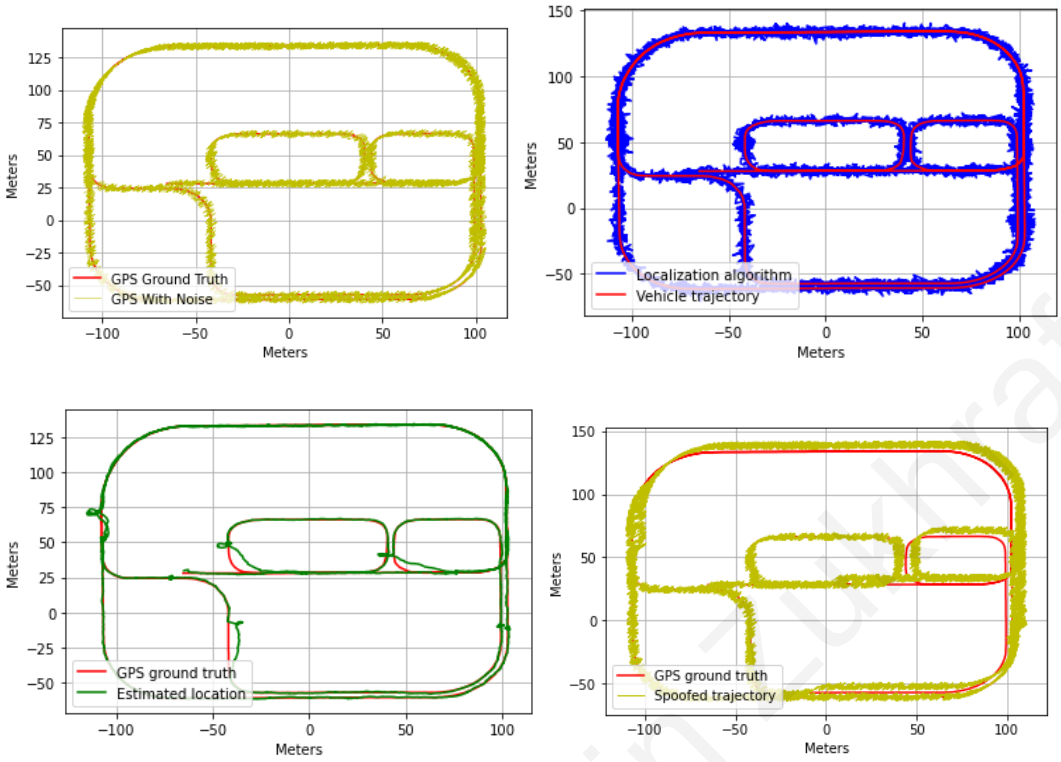


Figure 24. The above graphs show (a) GPS ground Truth, (b) the working of LA, (c) Estimated path, and (d) spoofed path for Map 10

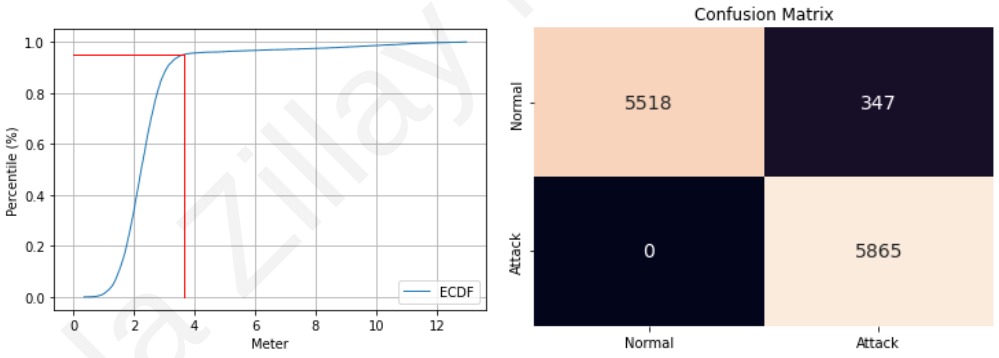


Figure 25. (a) shows the threshold on 95th percentile approx. equal to 3.8 and (b) shows the confusion matrix analysis

Trajectory 2:

Table 1. Parameter assignment for Map 2

	Parameter	**Value**
1	Length of data	5588
2	Attack bias	5

3	Sliding window	5
4	Standard Deviation of Signals of Opportunity	10
5	Standard Deviation of Signals of Opportunity Orientation	20
6	Standard Deviation of the Noise in the GPS	3
7	Sampling Interval in seconds	0.05
8	Percentile for threshold	95.0
9	Town name (From Carla)	Town 10

Graphs:

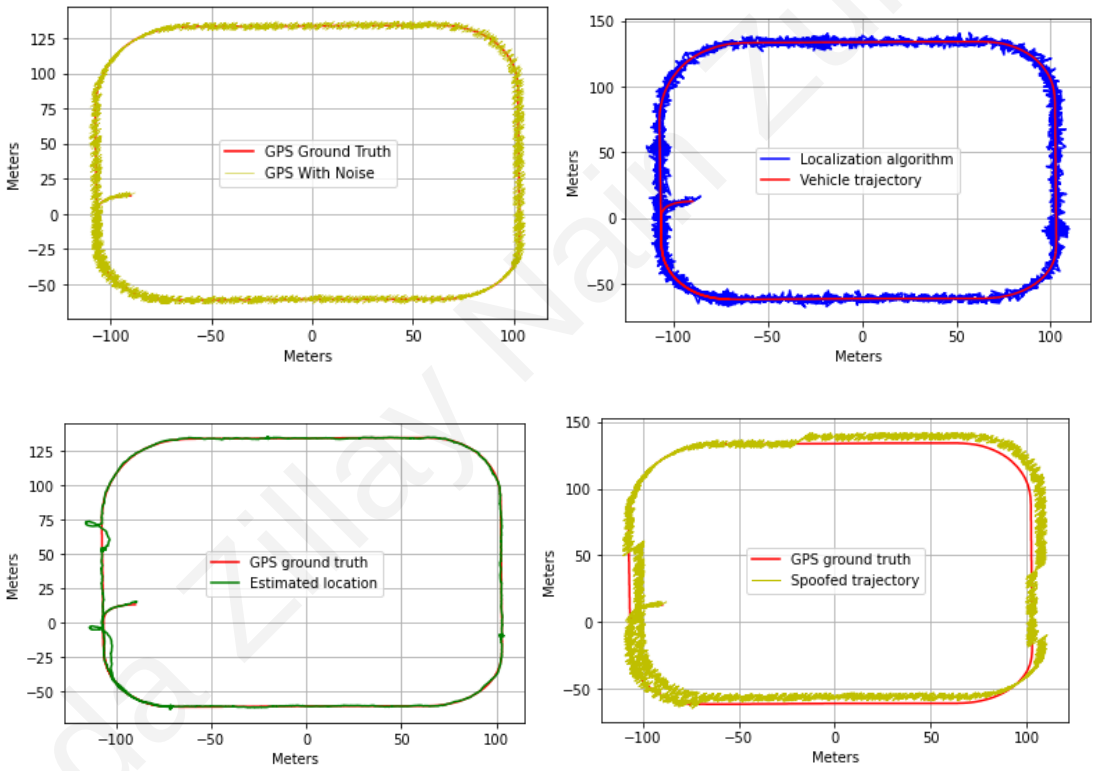


Figure 26. The above graphs show (a) GPS ground Truth, (b) the working of LA, (c) Estimated path, and (d) spoofed path for Map 10

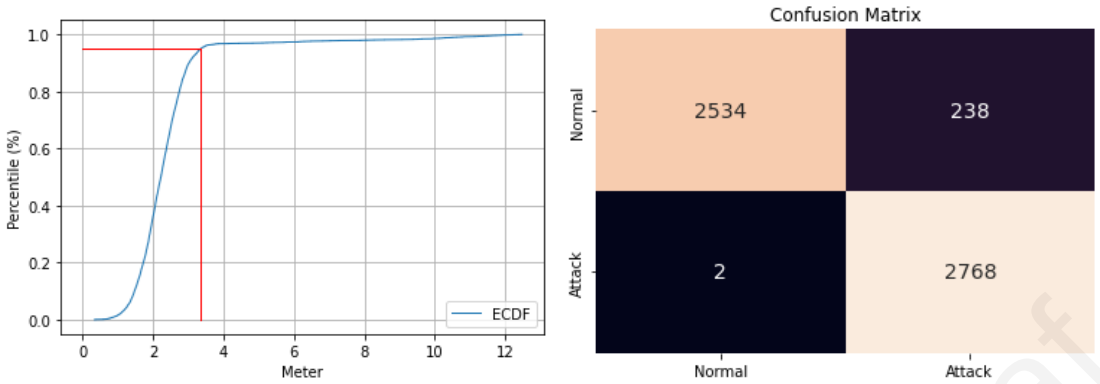


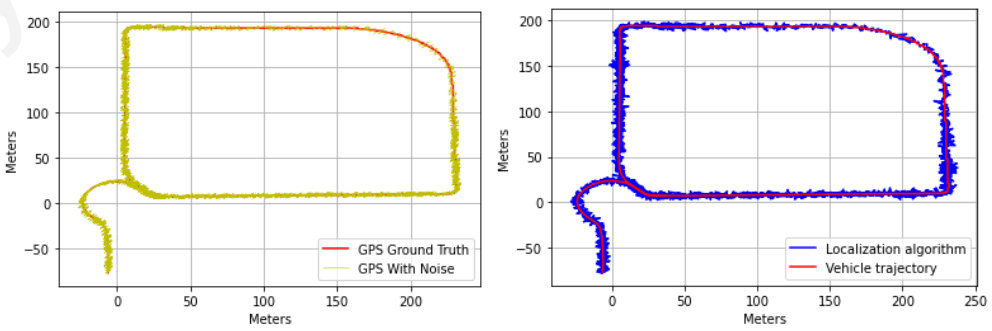
Figure 27. (a) shows the threshold on 95th percentile approx. equal to 3.6 and (b) shows the confusion matrix analysis

Trajectory 3:

Table 2. Parameter assignment for Map 3

	Parameter	**Value**
1	Length of data	4239
2	Attack bias	81
3	Sliding window	5
4	Standard Deviation of Signals of Opportunity	10
5	Standard Deviation of Signals of Opportunity Orientation	20
6	Standard Deviation of the Noise in the GPS	3
7	Sampling Interval in seconds	0.05
8	Percentile for threshold	95.0
9	Town name (From Carla)	Town 2

Graphs:



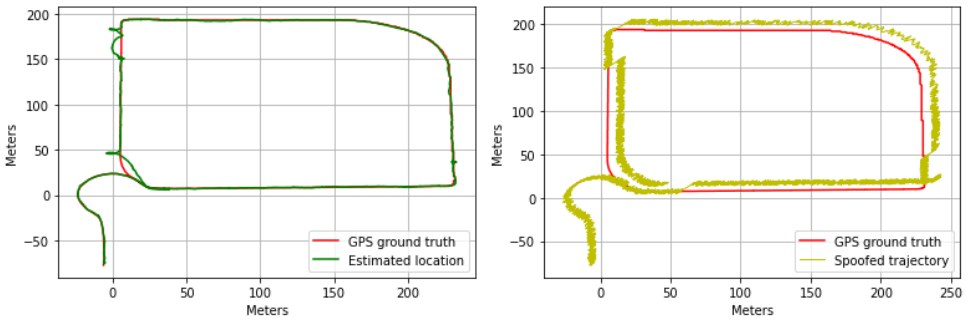


Figure 28. The above graphs show (a) GPS ground Truth, (b) the working of LA, (c) Estimated path, and (d) spoofed path for Map 10

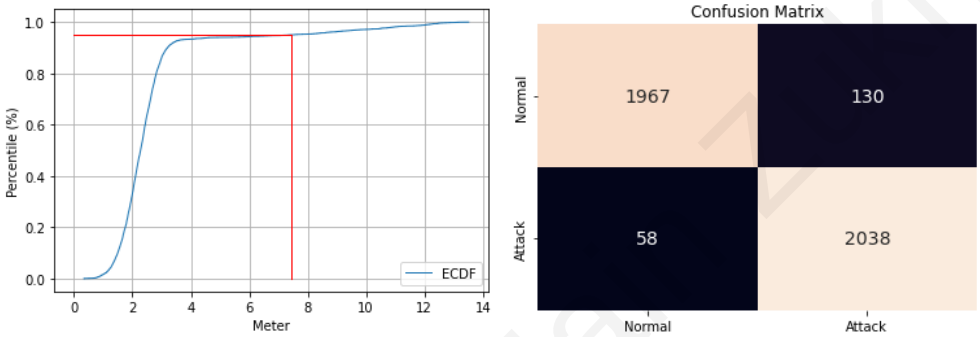


Figure 29. (a) shows the threshold on 95th percentile approx. equal to 6.7 and (b) shows the confusion matrix analysis

Trajectory 4:

Table 3. Parameter assignment for Map 4

	Parameter	**Value**
1	Length of data	9175
2	Attack bias	81
3	Sliding window	5
4	Standard Deviation of Signals of Opportunity	10
5	Standard Deviation of Signals of Opportunity Orientation	20
6	Standard Deviation of the Noise in the GPS	3
7	Sampling Interval in seconds	0.05
8	Percentile for threshold	95.0
9	Town name (From Carla)	Town 3

Graphs:

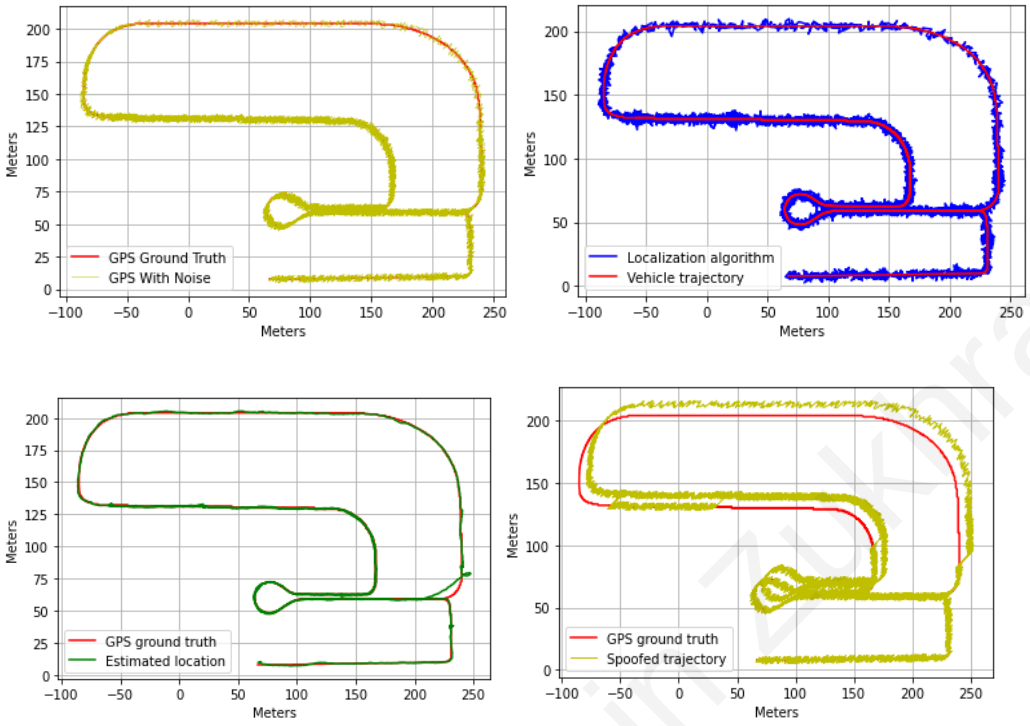


Figure 30. The above graphs show (a) GPS ground Truth, (b) the working of LA, (c) Estimated path, and (d) spoofed path for Map 10

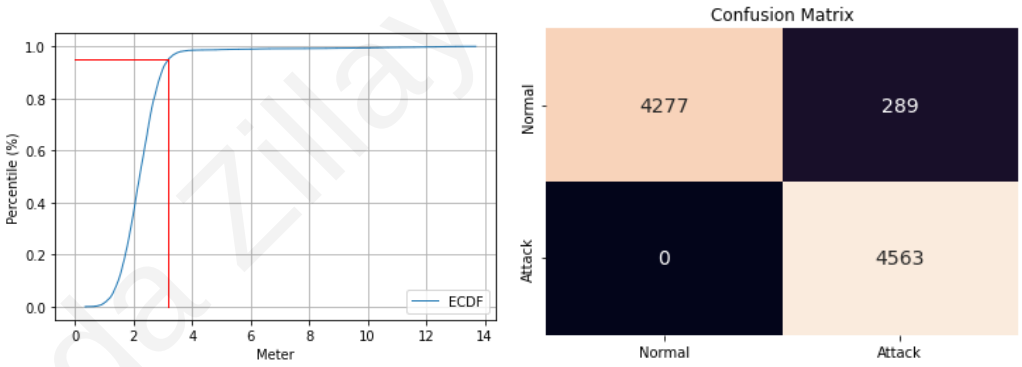


Figure 31. (a) shows the threshold on 95th percentile approx. equal to 2.6 and (b) shows the confusion matrix analysis

All these data sets, and test results were later used in chapter 5 to conduct intensive research and comparison by using ML algorithm for anomaly detection.

3.8 Robust Cooperative Sparse Representation Solutions for Detecting and Mitigating Spoofing Attacks in Autonomous Vehicles

In collaboration with the University of Patras team, we have combined our solutions and developed a robust approach for detecting spoofing attacks on the GPS of CAVs. Our specific contribution to the developed solution is discussed in this section. CAVs have the capability to

determine their own location using various sources of information. In addition to data collected from neighboring vehicles, they can utilize absolute location data obtained from wireless network infrastructure and relative location data based on previous estimations and in-vehicle inertial sensors. By incorporating a device equipped with SDR technology, the CAV can receive signals from the surrounding wireless infrastructure, such as cellular, Wi-Fi, or DVB-T, and extract essential information such as timing, angle, or signal strength from the corresponding transmitters. This allows the CAV to independently estimate its current position, bypassing the need for GPS measurements, through the utilization of known transmitter locations and a LA. The estimated position of the CAV is output by the device as a Gaussian random variable with a covariance matrix that incorporates the uncertainty in the predicted locations. These absolute positions can be refined using Bayesian filtering techniques such as the Extended Kalman Filter (EKF) and in-vehicle multi-source data fusion. During the Prediction phase, the CAV's location is projected forward in time using the prior location, inertial sensor measurements, and a mobility model. A Gaussian distribution governs the expected position at time instance. The EKF technique combines the projected location with the absolute position calculated by the LA during the Update phase, resulting in a revised location estimate presumed to be derived from a Gaussian distribution. The covariance matrix captures the revised location's uncertainty by using EKF algorithm mentioned in section 3.3.1. It is a part of the research “Robust Cooperative Sparse Representation Solutions for Detecting and Mitigating Spoofing Attacks in Autonomous Vehicles” accepted in MED 2023 Conference in Limassol, Cyprus as mentioned in the publication section. The Figure 32 below highlights (in orange color block) our contribution in the research.

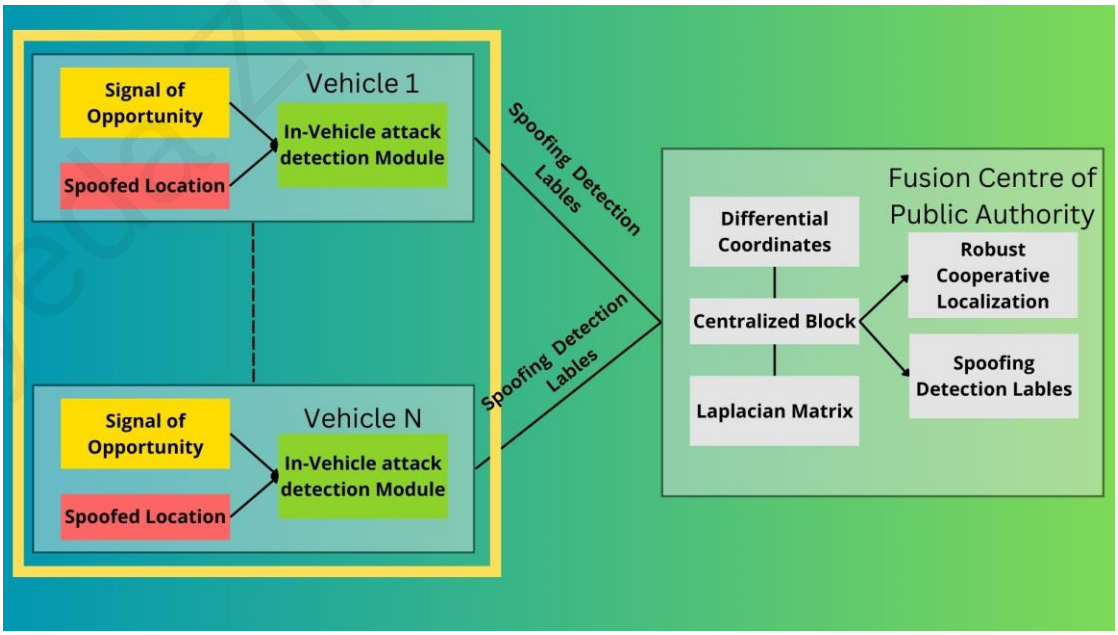


Figure 32. Conceptual architecture of robust cooperative sparse coding.

Chapter 4 - A Machine Learning Approach for Detecting GPS Location Spoofing Attacks in Autonomous Vehicles

4.1 Introduction

The text emphasizes the need of precise position data in the functioning of CAV. These location data are sent into the CAV's Advanced Driver Assistance System (ADAS) and perception engine, allowing it to perceive and interpret its surroundings. As a result, autonomous driving and navigation operations are supported, while accidents with neighboring cars and Vulnerable Road Users (VRUs) such as pedestrians and bicycles are avoided. Location awareness is also required for the deployment of VANET and the implementation of Vehicle-to-Vehicle/Infrastructure (V2V/V2I) standards, all of which are critical components of Intelligent Transportation Systems (ITS). Having accurate location information for CAVs is critical for improving ITS safety. Recent improvements, however, have focused attention on security risks to autonomous driving, raising worries about the dependability of location data and the steps required to assure its integrity. Addressing these challenges becomes critical in order to keep CAVs safe and reliable in ITS contexts [37-38].

As previously noted, attacks are classified into two types: signal processing techniques and data-driven tactics. Signal processing techniques examine GPS data and detect abnormalities that suggest a spoofing attack using specialized signal processing algorithms. For signal analysis, these approaches frequently necessitate the use of extra hardware and advanced algorithms. While they can be beneficial, they may not be practicable in many situations due to the increased resources required. Data-driven methods, on the other hand, make use of machine learning techniques to identify GPS spoofing attacks. These techniques examine GPS data for abnormalities that indicate spoofing. They learn from data rather than explicit knowledge of attack features or extra hardware. The demand for tagged data, which implies having GPS data that is precisely labeled as either normal or symptomatic of an attack, is one problem with data-driven techniques. In real-world circumstances, acquiring such labeled data, particularly attack data, might be problematic. Attacks may be difficult to replicate in controlled contexts for research, making it difficult to gather enough labeled data for training ML models. A data-driven strategy based on ML is suggested in this chapter for identifying spoofing attacks. The goal is to create a system that can detect spoofing attacks without the need of attack data or extra hardware. This technique provides a viable and effective solution for identifying GPS spoofing attempts by employing ML algorithms and assessing the features of GPS data [18][19].

4.2 Methodology

4.2.1 Problem Formulation

In this Chapter, we take an anomaly detection method to the problem of GPS location faking in autonomous vehicles. We denote the estimated location of the autonomous vehicle as $\mathrm{p}_k^L = [\mathrm{x}_k^L, \mathrm{y}_k^L]^T$ and the GPS location at time step k as $\mathrm{p}_k^G = [\mathrm{x}_k^G, \mathrm{y}_k^G]^T$. The positional difference between the two sites is represented by the differential feature dk R2, which we calculate to create features for analysis:

$$d_k = \mathrm{p}_k^G - \mathrm{p}_k^L = [\mathrm{x}_k^d, \mathrm{y}_k^d]$$

where $\mathrm{x}_k^d = \mathrm{x}_k^G - \mathrm{x}_k^L$ and $\mathrm{y}_k^d = \mathrm{y}_k^G - \mathrm{y}_k^L$.

We use a learning model (anomaly detector) termed f: $\mathrm{R}^2 \longrightarrow \{0,1\}$ to locate spoof locations. The model takes the input $d_k \in \mathrm{R}^2$ and predicts its label $L_k \in \{0,1\}$, denoted as $L_k = f(d_k)$, at each time step k. An attack-free site has a label value of 0, while a faked location has a value of 1. We present a sliding window strategy to capture the temporal characteristics of the data and allow for spoofing attempts that may have happened in earlier time steps. By taking into account the series of differentials such that $L_k = f(d_k, d_{k+1}, \ldots, d_{k-W})$, this sliding window of size W helps with the prediction task.

4.2.2 Attack detection Pipeline

As shown in Figure 34, the suggested method for identifying GPS position spoofing attacks follows a certain pipeline. There are three separate stages in the pipeline: training, validation, and testing.

An anomaly detector dubbed f is trained to recognize the typical behavior of the data during the Training stage. A training dataset, abbreviated as D_{tr}, is used to accomplish this. Attack-free areas, or situations without spoofing attacks, are only found in $D_{tr} = \{d_k\}_{k=1}^{|D_{tr}|}$.

Choosing the best learning model and its accompanying hyper-parameters is a step in the training process. A different dataset named D_{val} is used to do this. D_{val} is marked by the notation $|D_{val}| \ll |D_{tr},|$ because it is much smaller than the training set. In this dataset, attack-free locations and a smaller subset of attacked locations are represented as $D_{val} = \{(d_k, L_k)\}_{k=1}^{|D_{val}|}$.

On a separate test dataset, known as D_t, the testing stage is conducted. This dataset contains cases that the model hasn't seen in the training or validation sets, making it different from those sets. It can be described as $D_t = \{(d_k, L_k)\}_{k=1}^{|D_t|}$ and contains both attack-free and attacked

places. This stage's goal is to examine the trained model's performance on unobserved data, evaluating its propensity to correctly identify both attack-free and faked locations.

4.2.3 Learning Algorithm Pipeline

A total of seven datasets were used for training, validation, and testing. The initial step was to normalize the GPS data to a 0 to 1 scale. Following that, the training phase began, during which particular parameters were assigned to the model. Fitting these parameters to the training dataset and assessing their performance on the validation dataset were used to determine their trustworthiness. This iterative method was repeated until satisfactory results were obtained.

After fine-tuning the parameters with the validation dataset, they were applied to the testing dataset to acquire the final findings. This method was used for the learning algorithms, and the results were documented. Figure 33 shows a flow diagram describing this technique.

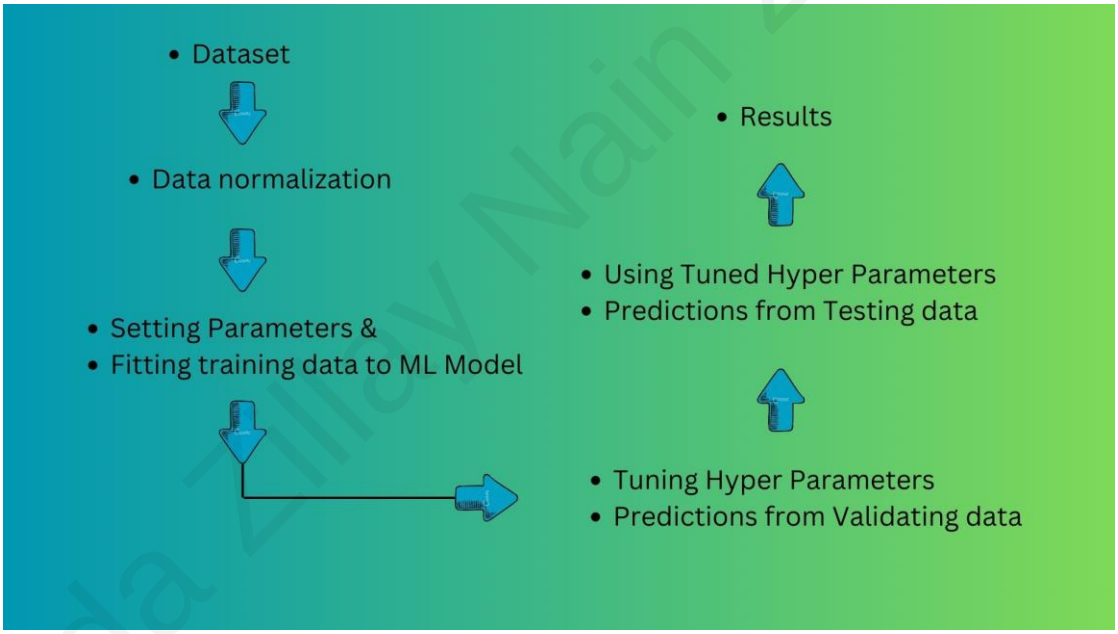


Figure 33. Flow diagram for Learning Algorithms.

4.3 Performance Evaluation

4.3.1 Experimental Setup

Same system specification is used as mentioned earlier in chapter 3 section 6.

We created a moving vehicle within the simulation environment during the Training stage, simulating typical driving conditions free of any threats. While including user-selected noise profiles for both the simulated GPS data and GPS-free estimated vehicle locations, which were produced based on cellular networks, the car followed a predetermined course. The untrained

Python Anomaly Detector was provided with the sensor data acquired during this phase utilizing ROS and a publish-subscribe mechanism. As shown in Figure 34, this procedure attempted to train the underlying ML model.

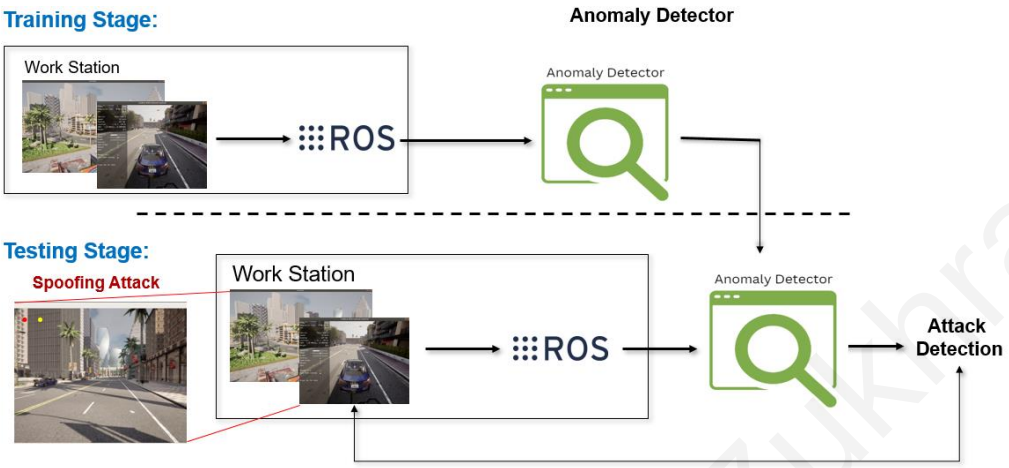


Figure 34. Experimental Setup.

During the testing phase, we created a moving vehicle that could travel in a variety of directions both normally and during an attack. We used noise profiles for the simulated GPS data and GPS-free estimated vehicle locations, similar to the Training stage. The CARLA simulator was then contacted to confirm the attack prediction labels, designated as L_k. A straightforward interface within CARLA was developed to depict two lights in the vehicle's cockpit, making visual verification easier. The right light, which remained off in normal circumstances and turned yellow when an attack was underway, served as the primary indicator of the attack status. When no attacks were found, the left light displayed green information; when attacks were found, it displayed red information. Even while simulation data and attack prediction labels may be gathered for offline study, the CARLA interface made it simple to compare the real-time effectiveness of different attack detection techniques.

4.3.2 Simulation Parameters and datasets

By inserting various attack biases b into the GPS measurements throughout our simulation, we looked into several test situations. These attack biases changed depending on the vehicle's trajectory. Nevertheless, a few variables remained the same in every test instance. These variables were $\sigma^L = 10m$ (which represented the standard deviation of predicted vehicle locations calculated without the use of a GPS receiver), $\sigma^G = 10m$ (which represented the standard deviation of GPS measurements), and a sample interval of $\Delta t = 0.05s$.

We produced a total of seven trajectories, represented by the letters T_i, where i is a number between 1 and 7. The accompanying datasets are compiled in TABLE 4. Notably, the attack bias was particularly modelled for trajectories T_6 and T_7 with different values.

Table 4. Number of total, normal, and attacked data points in each trajectory.

No	Total	Normal	Attacked	Attack Bias [m]
T_1	6,020	3,303	2,718	5
T_2	13,433	13,433	0	0
T_3	10,265	10,265	0	0
T_4	11,730	11,730	0	0
T_5	4,542	2,272	2,270	5
T_6	4,193	2,097	2,096	5,6,9
T_7	9,129	4,566	4,563	5,9

The train set, the validation set, and the test set were created from the acquired data. There were no faked or attacked datasets in the train set, which consisted of three normal datasets reflecting the trajectories T_2, T_3 and T_4. The learning model was trained exclusively using regular data, without any GPS spoofing attacks, because the problem was defined as an anomaly detection task.

Two "spoof" datasets representing the trajectories T_1 and T_5 made up the validation set, which had an attack bias of $b = 5$. The test set, however, was made up of five "faked" datasets that represented the trajectories T_6 and T_7. The biases experienced by each model during training ($b = 5$) which was done indirectly through the validation set—were not the same as the attack biases in the test set.

4.3.3 Simulation Results

First, taken into consideration sliding window ($w = 10$) we investigate how the machine learning (ML)-based anomaly detection approaches that have been put to the test performs. Performance is noticeably worse when no prior data is taken into consideration ($w = 0$) than when different window sizes are examined. The results show that performance improves up to a certain degree when the window size grows. After that, the performance starts to deteriorate.

This happens as a result of short-term temporal correlations between successive data points being considered, which helps prevent inaccurate oscillations between 'attack' and 'normal'

detection. However, when the ground truth alternates between "normal" and "attack," considering a large number of prior data points loses some of its effectiveness. The iForest, LOF, OC-SVM, AE, VAE, and DBN models ideal window sizes are discovered to be 5, 10, 10, 20, 10, and 10 respectively. The outcomes for the iForest and AE models are averaged across 50 iterations in order to address the stochastic character of the methods.

4.3.4 Role of Learning Models

In this section, we examine the effectiveness of several ML-based outlier identification techniques while considering a window size of 10, which showed the best performance in the section before this one.

With a Gm (Geometric mean) score of 97.85% and an F1 score of 97.84%, the LOF (Local Outlier Factor) model had the best performance among the ML-based models tested. The Gm and F1 scores for the AE algorithm were 94.34% and 93.96%, respectively, making it the algorithm with the poorest performance. The Gm and F1 scores for the OC-SVM model were 97.36% and 97.33%, respectively, while the Gm and F1 scores for the iForest model were 97.60% and 97.63%, respectively. We chose the top-performing ML-based method, LOF, to compare with the data-driven TAD (Threshold Anomaly Detection) method for identifying location spoofing attacks. To guarantee an equal amount of data for testing, we altered the TAD data by eliminating the first 10 data points.

Table 5. Comparative analysis of different Learning Models

Model	Gm	F1
$OC-SVM$	97.36	97.33
$iForest$	97.60	97.63
LOF	97.85	97.84
AE	95.64	95.39
VAE	96.95	96.91
DBN	96.70	96.66
TAD	83.49	83.92

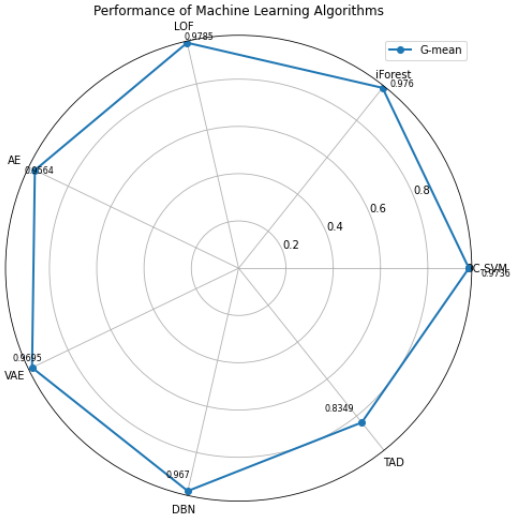


Figure 35. Radar map for G-Mean

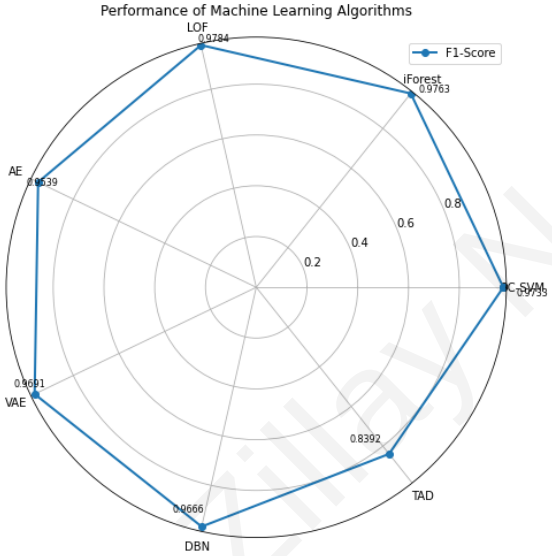


Figure 36. Radar map for F1 score

The findings, show that LOF performs better than TAD in all performance criteria. On the test set, specifically, LOF received a Gm score of 98.43% and an F1 score of 98.45%, which is specifically 15% higher than TAD's Gm score of 83.49% and F1 score of 83.92%. Figure and Figure 35 and 36 shows the radar map plot of G-mean and F1 score whereas Figure 37 shows the heat map.

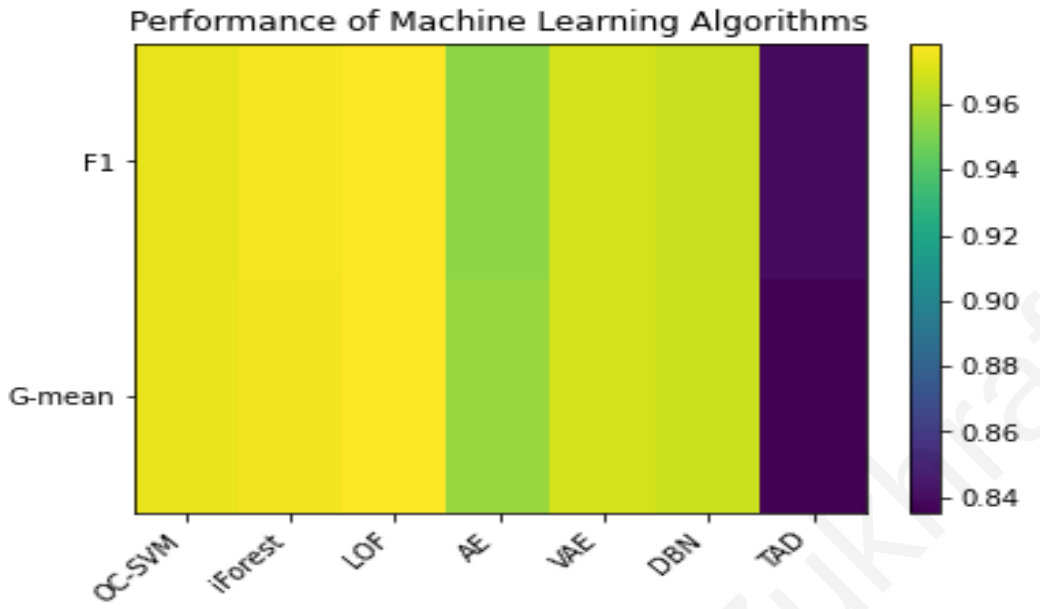


Figure 37. Heat map of the results

Chapter 5 - Conclusion and Future Works

Finally, the GPS spoofing detection method created in this thesis performed well throughout simulation testing. In chapter 4, we discuss the problem of GPS position spoofing attacks when autonomous vehicles are in motion in this work and suggest an ML-based anomaly detection approach to spot such attempts. Our method stands out because it delivers outstanding classification accuracy with F1-scores and G-means ranging from 95% to 98% depending on the ML model employed, and it does so without requiring attack data during the training stage. Notably, among the ML-based strategies examined, our solution based on the LOF model outperformed another state-of-the-art non-ML-based solution by 15% in terms of detection accuracy. For Future work I would suggest the following research:

1. As part of our ongoing research and as future works at the KIOS Centre of Excellence, University of Cyprus, we intend to deploy the suggested solution in Chapter 3 on one of our connected autonomous vehicles (CAVs) and implement it on an embedded computer device Nvidia Jetson Nano. This will allow us to test the approach in actual situations where the vehicle is moving and GPS location spoofing attacks are carried out using open-source spoofing software and commercial off-the-shelf (COTS) SDR gear.
2. Additionally, we want to look into online ML-based algorithms that can change in response to changing circumstances, such the fact that open-sky rural places have lower GPS location uncertainty than cities. Even under different environmental conditions, we want to make sure that detection accuracy is good and false positives are kept to a minimum. Both of aforementioned algorithms are expected to be tested in real environment because right now we are dealing with specific noise profiles and in reality, the scenario is different.
3. The GPS spoofing detection solution's performance was assessed in several scenarios to determine its resilience and flexibility. There were three separate habitats considered: rural, suburban, and urban as shown in Figure 38. The system proved its capacity to identify GPS spoofing with relatively low noise levels in the rural setting, where there are normally less barriers and less interference. Moving the system to a suburban area with modest barriers and interference, it was evaluated to guarantee its efficiency in detecting spoofing attacks while compensating for increased noise in GPS readings. The system was evaluated in an urban environment with high-rise buildings and extensive infrastructure for its ability to identify GPS spoofing despite severe signal obstacles and greater levels of noise in GPS reading. By assessing the solution's performance in these many contexts, a full knowledge of its flexibility and efficacy in

various real-world circumstances may be achieved. The results for this problem are shown in Figure 38.

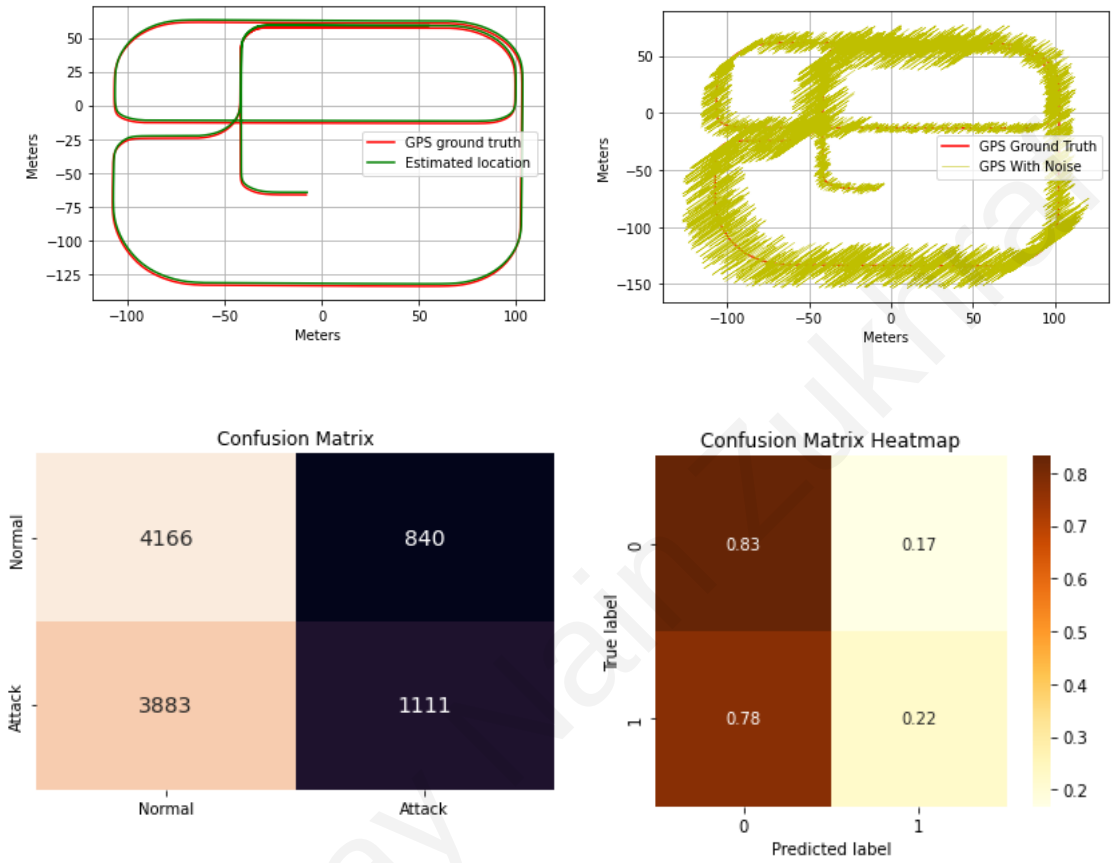


Figure 38. (a)estimated state and ground truth of vehicle (b) different noise profiles, (c) confusion matrix, (d) heat map.

From the Figure 38 we can clearly see how the Attack Detection solution discussed in Chapter 3 works. Different noise profile magnitude can be observed in Figure 38(b) where as from confusion matrix in Figure 38(c) it clearly shows that some work needs to be done.

References

[1] X. Sun, F. R. Yu and P. Zhang, "A Survey on Cyber-Security of Connected and Autonomous Vehicles (CAVs)," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 7, pp. 6240-6259, July 2022, doi: 10.1109/TITS.2021.3085297.

[2] M. Singh, "Cybersecurity in vehicular communication" in Information Security of Intelligent Vehicles Communication, Springer, 2021.

[3] R. K. Jaiswal and C. Jaidhar, "A performance evaluation of location prediction position-based routing using real gps traces for vanet", Wireless Personal Communications, vol. 102, no. 1, pp. 275-292, 2018.

[4] K. Ren, Q. Wang, C. Wang, Z. Qin and X. Lin, "The security of autonomous driving: Threats defenses and future directions", Proceedings of the IEEE, vol. 108, no. 2, pp. 357-372, 2019.

[5] R. Ferreira, J. Gaspar, P. Sebastião and N. Souto, "Effective gps jamming techniques for uavs using low-cost sdr platforms", Wireless Personal Communications, vol. 115, no. 4, pp. 2705-2727, 2020.

[6] S. Hussein, A. Krings and A. Azadmanesh, "Vanet clock synchronization for resilient dsrc safety applications", 2017 Resilience Week (RWS), pp. 57-63, 2017.

[7] T. Infrastructure, Vulnerability assessment of the transportation infrastructure relying on the global positioning system, 2001.

[8] A. Neri, C. Stallo, A. Coluccia, V. Palma, P. Salvatori, A. Vennarini, O. Pozzobon, G. Gamba, S. Fantinato, M. Barbuto et al., "An anti-jamming and anti-spoofing digital beamforming platform for the gnss-based ertms train control system", Proceedings of the 30th International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2017), pp. 3538-3556, 2017.

[9] R. Xu, M. Ding, Y. Qi, S. Yue and J. Liu, "Performance analysis of gnss/ins loosely coupled integration systems under spoofing attacks", Sensors, vol. 18, no. 12, pp. 4108, 2018.

[10] E. Ranyal and K. Jain, "Unmanned aerial vehicle's vulnerability to gps spoofing a review", Journal of the Indian Society of Remote Sensing, pp. 1-7, 2020.

[11] Y.-C. Liu, G. Bianchin and F. Pasqualetti, "Secure trajectory planning against undetectable spoofing attacks", Automatica, vol. 112, pp. 108655, 2020.

[12] L. He, H. Li and M. Lu, "A fundamental architecture of anti-spoofing gnss receiver", China Satellite Navigation Conference, pp. 899-909, 2017.

[13] M. A. Hossain, I. Elshafiey and A. Al-Sanie, "Cooperative vehicle positioning with multi-sensor data fusion and vehicular communications", Wireless Networks, vol. 25, no. 3, pp. 1403-1413, 2019.

[14] C. Vitale, N. Piperigkos, C. Laoudias, G. Ellinas, J. Casademont, J. Escrig, A. Kloukiniotis, A. S. Lalos, K. Moustakas, R. D. Rodriguez et al., "Caramel: results on a secure architecture for connected and autonomous vehicles detecting gps spoofing attacks", EURASIP Journal on Wireless Communications and Networking, vol. 2021, no. 1, pp. 1-28, 2021.

[15] F. L. Lobo, D. C. Grael, H. A. d. Oliveira, L. A. Villas, A. Almehmadi and K. El-Khatib, "A distance-based data fusion technique for minimizing gps positioning error in vehicular ad hoc networks", Proceedings of the 15th ACM International Symposium on QoS and Security for Wireless and Mobile Networks, pp. 101-108, 2019.

[16] K.-C. Kwon and D.-S. Shim, "Performance analysis of direct gps spoofing detection method with ahrs/accelerometer", Sensors, vol. 20, no. 4, pp. 954, 2020.

[17] J. Shen, J. Y. Won, Z. Chen and Q. A. Chen, "Drift with devil: Security of multi-sensor fusion based localization in high-level autonomous driving under {GPS} spoofing", 29th {USENIX} Security Symposium ({USENIX} Security 20)2020, pp. 931-948.

[18] J. Friedt, W. Feng, D. Rabus and G. Goavec-Merou, "Real time gnss spoofing detection and cancellation on embedded systems using software defined radio", EuCAP Dusseldorf Germany 2021.

[19] F. A. Milaat and H. Liu, "Decentralized detection of gps spoofing in vehicular ad hoc networks", IEEE Communications Letters, vol. 22, no. 6, pp. 1256-1259, 2018.

[20] S. So, P. Sharma, and J. Petit, "Integrating plausibility checks and machine learning for misbehavior detection in VANET," in Proc. 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018, pp. 564–571.

[21] M. R. Manesh, J. Kenney, W. C. Hu, V. K. Devabhaktuni, and N. Kaabouch, "Detection of GPS spoofing attacks on unmanned aerial systems," in Proc. 16th IEEE Annual Consumer Communications & Networking Conference (CCNC), 2019, pp. 1–6.

[22] Y. Arjoune, F. Salahdine, M. S. Islam, E. Ghribi, and N. Kaabouch, "A novel jamming attacks detection approach based on machine learning for wireless communication," in Proc. IEEE International Conference on Information Networking (ICOIN), 2020, pp. 459–464.

[23] E. Shafiee, M. R. Mosavi, and M. Moazedi, "Detection of spoofing attack using machine learning based on multi-layer neural network in single-frequency GPS receivers," The Journal of Navigation, vol. 71, no. 1, pp. 169–188, 2018.

[24] K. H. Park, E. Park, and H. K. Kim, "Unsupervised fault detection on unmanned aerial vehicles: Encoding and thresholding approach," Sensors, vol. 21, no. 6, pp. 1–17, 2021.

[25] C. Ryan, F. Murphy, and M. Mullins, "Semiautonomous vehicle risk analysis: A telematics-based anomaly detection approach," Risk analysis vol. 39, no. 5, pp. 1125–1140, 2019.

[26] M. Said Elsayed, N.-A. Le-Khac, S. Dev, and A. D. Jurcut, "Network anomaly detection using LSTM based autoencoder," in Proc. 16th ACM Symposium on QoS and Security for Wireless and Mobile Networks, 2020, pp. 37–45.

[27] M. Kamal, A. Barua, C. Vitale, C. Laoudias and G. Ellinas, "GPS Location Spoofing Attack Detection for Enhancing the Security of Autonomous Vehicles," 2021 IEEE 94th Vehicular Technology Conference (VTC2021-Fall), Norman, OK, USA, 2021, pp. 1-7, doi: 10.1109/VTC2021-Fall52928.2021.9625567.

[28] C. Gao et al., "Autonomous driving security: State of the art and challenges," IEEE Internet of Things Journal, vol. 9, no. 10, pp. 7572–7595, 2022.

[29] A. Kloukiniotis et al., "Countering adversarial attacks on autonomous vehicles using denoising techniques: A review," IEEE Open Journal of Intelligent Transportation Systems, vol. 3, pp. 61–80, 2022.

[30] R. T. Ioannides, T. Pany, and G. Gibbons, "Known vulnerabilities of global navigation satellite systems, status, and potential mitigation techniques," Proceedings of the IEEE, vol. 104, no. 6, pp. 1174–1194, 2016.

[31] D. Schmidt et al., "A survey and analysis of the gnss spoofing threat and countermeasures," ACM Comput. Surv., vol. 48, no. 4, may 2016. [Online]. Available: https://doi.org/10.1145/2897166

[32] Z. Zhang, L. Zhou, and P. Tokekar, "Strategies to design signals to spoof kalman filter," in 2018 Annual American Control Conference (ACC), 2018, pp. 5837–5842.

[33] J. Su, J. He, P. Cheng, and J. Chen, "A stealthy gps spoofing strategy for manipulating the trajectory of an unmanned aerial vehicle," IFACPapersOnLine, vol. 49, no. 22, pp. 291–296, 2016, 6th IFAC Workshop on Distributed Estimation and Control in Networked Systems NECSYS 2016.

[34] D. Medina, H. Li, J. Vil`a-Valls, and P. Closas, “On robust statistics for gnss single point positioning,” in 2019 IEEE Intelligent Transportation Systems Conference (ITSC), 2019, pp. 3281–3287.

[35] F. Faurie and A. Giremus, “Combining generalized likelihood ratio and m-estimation for the detection/compensation of gps measurement biases,” in 2010 IEEE International Conference on Acoustics, Speech and Signal Processing, 2010, pp. 4178–4181.

[36] D. Mori, H. Sugiura, and Y. Hattori, “Adaptive sensor fault detection and isolation using unscented kalman filter for vehicle positioning,” in 2019 IEEE Intelligent Transportation Systems Conference (ITSC), 2019, pp. 1298–1304.

[37] K. Ren, Q. Wang, C. Wang, Z. Qin, and X. Lin, “The security of autonomous driving: Threats, defenses, and future directions,” Proceedings of the IEEE, vol. 108, no. 2, pp. 357–372, 2019.

[38] M. L. Psiaki and T. E. Humphreys, “GNSS spoofing and detection,” Proceedings of the IEEE, vol. 104, no. 6, pp. 1258–1270, 2016.

[39] S. Rezaei and R. Sengupta, "Kalman filter-based integration of dgps and vehicle sensors for localization", IEEE Transactions on Control Systems Technology, vol. 15, no. 6, pp. 1080-1088, 2007.

[40] IMU: http://wiki.ros.org/razor_imu_9dof

[41] VESC: https://github.com/f1tenth/vesc_firmware

[42] GNSS:

https://content.u-blox.com/sites/default/files/C94-M8PAppBoard_UserGuide_%28UBX-15031066%29.pdf

[43] NASA Earth data:

https://www.earthdata.nasa.gov/eosdis/science-system-description/eosdis-components/earthdata-login

Appendix

Appendix A: IMU Sensor

To calibrate [40] the IMU, perform the following steps:

1. Get the Arduino IDE, an integrated development environment (IDE) for programming Arduino boards.
2. Follow these instructions to install the SparkFun Board Add-on in the Arduino IDE:
    a. Launch the Arduino IDE, then navigate to File > Preferences.
    b. Locate the "Additional Board Manager URLs" field in the Preferences window and click the icon next to it.
    c. Enter the following URL for the SparkFun Board Add-on: "https://raw.githubusercontent.com/sparkfun/Arduino_Boards/main/IDE_Board_Manager/package_sparkfun_index.json$ catkin_make"
3. Install the SparkFun Apollo3 Boards in the Arduino IDE by navigating to Tools > Board > Board Manager and searching for "SparkFun Apollo3 Boards." When the board package appears in the list, install it.
4. In the Arduino IDE, choose the SparkFun RedBoard Artemis ATP as the board type.
5. In the Arduino IDE, open the file "src/Razor_AHRS/Razor_AHRS".
6. Import the SparkFun Library for the IMU by doing the following steps:
    a. In the Arduino IDE, navigate to the library manager by selecting "Sketch" > "Include Library" > "Manage Libraries."
    b. In the Library Manager, search for "SparkFun_ICM_20948_IMU" and install the library by clicking the "Install" butt.

Appendix B: VESC Sensor

To calibrate [41] the VESC, perform the following steps:

1. Begin by making sure the VESC board has the correct firmware file ("VESC_60_MkV_5.02_SERVO_OUT.bin"). This file may be found in the main directory. *Important: Connect a fully charged battery to the VESC driver before beginning.*
2. Connect the battery to the VESC driver and wait for the blue LED to stabilize.
3. Connect your VESC board to your computer via the USB cable.
4. Launch the VESC Tool, which is software for configuring the VESC's firmware.
5. To connect to the VESC board, click the "Autoconnect" button in the program.
6. Navigate to the "Firmware" area on the left side of the program interface.

7. Navigate to the "Custom File" tab inside the "Firmware" section.
8. Click on the folder icon to get to the VESC_TOOL folder, which includes the firmware binary file for your specific VESC board.
9. After you've chosen the firmware binary file, press the "Upload" button in the bottom-right corner of the program interface.
10. Wait at least 10 seconds after the firmware upload is complete before reconnecting to the VESC board to specify further parameters.