Multimodal In-Vehicle Sensor Fusion for Cyber-Secured Autonomous Navigation

Final Report

by

Mizanur Rahman, *The University of Alabama*
Sagar Dasgupta, *The University of Alabama*
Muhammad Sami Irfan, *The University of Alabama*
Mashrur Chowdhury, *Clemson University*
Long Cheng, *Clemson University*
M Sabbir Salek, *Clemson University*

February 2025

NATIONAL CENTER FOR TRANSPORTATION CYBERSECURITY AND RESILIENCY (TraCR)

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The report is funded, partially or entirely, by the National Center for Transportation Cybersecurity and Resiliency (TraCR) under Grant No. 69A3552344812 and 69A3552348317 which is headquartered at Clemson University, South Carolina, USA, from the U.S. Department of Transportation's University Transportation Centers Program. The U.S. Government assumes no liability for the contents or use thereof.

CONTACTS

For more information:

Mizanur Rahman
2007 SCIB, 28 Kirkbride Lane,
Tuscaloosa, AL 35487-0288
Phone: (205) 348-1717
Email: mizan.rahman@ua.edu

TraCR
Clemson University
One Research Dr
Greenville, SC 29607
tracr@clemson.edu

ACKNOWLEDGMENT

The study is funded by the National Center for Transportation Cybersecurity and Resiliency (TraCR) under Grant No. 69A3552344812 and 69A3552348317, which is headquartered at Clemson University, South Carolina, USA, from the U.S. Department of Transportation's University Transportation Centers Program. This study is also partially supported by grants from the National Science Foundation (Award # 2104999 and Award # 2340456). The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. The U.S. Government assumes no liability for the contents or use thereof.

Technical Report Documentation Page

1. Report No. 10	**2. Government Accession No.** N/A	**3. Recipient's Catalog No.** N/A
4. Title and Subtitle Multimodal In-Vehicle Sensor Fusion for Cyber-Secured Autonomous Navigation		**5. Report Date:** February 2025 **6. Performing Organization Code:** N/A
7. Author(s) Mizanur Rahman, Ph.D.; https://orcid.org/0000-0003-1128-753X Sagar Dasgupta, Ph.D.; https://orcid.org/0000-0001-8491-662X Muhammad Sami Irfan; https://orcid.org/0000-0002-1116-935X Mashrur Chowdhury, Ph.D.; https://orcid.org/0000-0002-3275-6983 Long Cheng, Ph.D.; https://orcid.org/0000-0003-1736-0873 M Sabbir Salek, Ph.D.; https://orcid.org/0000-0001-7326-3694		**8. Performing Organization Report No.** 10
9. Performing Organization Name and Address National Center for Transportation Cybersecurity and Resiliency (TraCR), Clemson University, 414 A One Research Dr, Greenville, SC 29607 Department of Civil, Construction & Environmental Engineering, The University of Alabama 2007, Smart Communities and Innovation Building (SCIB) 28 Kirkbride Lane, Tuscaloosa, Box 870288, AL 35487-0288		**10. Work Unit No.** N/A **11. Contract or Grant No.** 69A3552344812 and 69A3552348317
12. Sponsoring Agency Name and Address U.S. Department of Transportation, Office of the Assistant Secretary for Research and Technology, 1200 New Jersey Avenue, SE, Washington, DC 20590		**13. Type of Report and Period Covered** Final Report, 01/01/2024 - 12/31/2024 **14. Sponsoring Agency Code** OST-R
15. Supplementary Notes Conducted under the U.S. DOT Office of the Assistant Secretary for Research and Technology's (OST-R) University Transportation Centers (UTC) program.		

16. Abstract

The security and reliability of Global Navigation Satellite Systems (GNSS) are critical for the safe operation of autonomous vehicles (AVs). However, GNSS signals are vulnerable to spoofing attacks, which can mislead AV navigation systems and compromise operational safety. This report addresses this vulnerability by developing detection and mitigation against GNSS spoofing attacks, with a particular focus on a slow drift GNSS spoofing attack. The study first investigates slow drift GNSS spoofing attacks, a stealthy form of signal manipulation where the attack gradually shifts the perceived position of the AV without triggering traditional anomaly detection mechanisms. To counteract this type of attack, a multi-sensor fusion-based spoofing detection framework is presented, integrating data from in-vehicle sensors, including speedometers, accelerometers, and gyroscope sensors, to cross-validate GNSS-derived positioning. The framework utilizes Long Short-Term Memory (LSTM) neural networks to predict vehicle motion and detect inconsistencies indicative of spoofing attacks. Field experiments validate the framework's effectiveness, achieving a 97.3% accuracy in identifying spoofing scenarios while maintaining a low false positive rate. Additionally, a Geographical Information Systems (GIS) and landmark-based navigation approach is developed for AV localization in GNSS-contested environments. This approach leverages in-vehicle sensor data, 2D map, and urban landmarks, such as roadway intersection, to provide an alternative positioning method. LSTM and Kalman Filters are utilized to enhance location shift prediction and position correction. The results of this research underscore the necessity of integrating multi-sensor validation and GIS-assisted navigation into AV systems to mitigate GNSS spoofing threats. The approaches could contribute to the development of cyber-resilient AV navigation frameworks, enhancing security in urban and contested environments in the presence of GNSS spoofing threats.

17. Keywords Global Positioning System (GPS), Autonomous Vehicles, Cybersecurity, and Urban Areas		**18. Distribution Statement** No restrictions.	
19. Security Classif. (of this report) Unclassified	**20. Security Classif. (of this page)** Unclassified	**21. No. of Pages** 68	**22. Price** N/A

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

TABLE OF CONTENTS

List of Tables

List of Figures

EXECUTIVE SUMMARY

Autonomous vehicles (AVs) rely on Global Navigation Satellite System (GNSS) signals for localization and navigation. However, GNSS spoofing attacks pose a critical cybersecurity threat, capable of manipulating satellite signals to mislead AVs, thereby jeopardizing passenger safety and operational reliability. This report addresses this vulnerability by developing detection and mitigation against GNSS spoofing attacks, with a particular focus on a slow drift GNSS spoofing attack. This study first investigates slow drift GNSS spoofing attacks, a stealthy form of signal manipulation where the attack gradually shifts the perceived position of the AV without triggering traditional anomaly detection mechanisms. The attack model successfully replicated satellite reception patterns and altered the vehicle's perceived location over time, even during turns, without immediate detection by conventional GNSS security mechanisms. Field experiments validated that slow-drift spoofing can mislead AVs without triggering alarms, confirming the necessity of robust detection and mitigation strategies. To counteract this type of attack, a multi-sensor fusion-based spoofing detection framework is presented, integrating data from in-vehicle sensors, including speedometers, accelerometers, and gyroscope sensors, to cross-validate GNSS-derived positioning. The framework utilizes Long Short-Term Memory (LSTM) neural networks to predict vehicle motion and detect inconsistencies indicative of spoofing attacks.

To counteract these threats, a sensor fusion-based GNSS spoofing attack detection framework was designed and rigorously tested. The framework integrates real-time kinematic (RTK) GNSS data with high-frequency inertial measurement unit (IMU) sensor readings, enabling anomaly detection based on inconsistencies between expected and actual motion. Two detection strategies were implemented: (1) an LSTM neural network for predicting vehicle movement based on in-vehicle sensor data and comparing it with GNSS-derived location shifts, and (2) a Random Forest (RF) machine learning model for detecting and classifying turns based on gyroscope data. The results confirm that the LSTM model achieved a 97.3% accuracy in detecting spoofing-induced location anomalies, while the RF model demonstrated 95.6% accuracy in turn classification and anomaly detection. These results highlight the effectiveness of machine learning-driven sensor fusion approaches in detecting GNSS spoofing attacks and improving AV cybersecurity.

Beyond attack detection, this study presents a GIS and landmark-based navigation system as a resilient alternative for AVs operating in GNSS-compromised environments. The framework combines IMU data, speedometer readings, and real-time intersection detection through computer vision, ensuring continuous localization without GNSS dependence. A Kalman Filter (KF) was used for position correction, while an LSTM-based model estimated travel distances to enhance navigation accuracy. Field testing in urban environments where GNSS signals were intentionally degraded confirmed that landmark-aided navigation significantly improved localization accuracy, reducing position errors by 85% compared to GNSS-only navigation. Additionally, the KF approach outperformed the LSTM model in distance prediction, lowering mean localization errors and improving real-time position estimates. These results confirm the feasibility of GIS-assisted navigation as a backup system, demonstrating its ability to maintain reliable AV operation in GNSS-denied environments.

This research successfully develops and validates a comprehensive GNSS security framework, integrating attack modeling, detection, and alternative navigation solutions to safeguard AV navigation against spoofing threats. The stealthy slow-drift spoofing attack model highlights the real-world risks of cyber manipulation, the sensor fusion-based detection framework provides a high-accuracy defense mechanism, and the GIS and landmark-based navigation system ensures continued AV operation in urban environments with GNSS limitations. Together, these findings establish an effective, multi-layered cybersecurity approach to fortify AV navigation systems against emerging GNSS threats.

While this study demonstrates significant advancements in GNSS attack modeling, detection, and mitigation, further research is necessary to enhance the real-time adaptation of spoofing detection algorithms and to expand the dataset with more diverse attack scenarios. Future work should also explore additional sensor modalities, such as LiDAR-based object recognition and radar-based motion tracking, to further improve AV resilience. As AV adoption continues to grow, these developments will be critical in ensuring the safe, secure, and reliable operation of autonomous transportation systems in both GNSS-reliable and GNSS-contested environments.

CHAPTER 1

Introduction

The future of transportation is autonomous vehicles (AVs), a paradigm shift that promises to revolutionize mobility by enhancing safety, efficiency, and accessibility. AVs operate through a sophisticated integration of sensors, actuators, and intelligent control systems. At the heart of AV operations lies the Global Navigation Satellite System (GNSS), which provides the real-time positioning and navigational data essential for autonomous functionality. However, this heavy reliance on GNSS also introduces significant vulnerabilities. GNSS signals are susceptible to unintentional disturbances, such as atmospheric anomalies and multipath propagation in urban environments (Spilker, 1996; Spilker Jr *et al.*, 1996), as well as to deliberate cyber-attacks like jamming and spoofing(Zeng *et al.*, 2017, 2018; Zidan, Elijah I Adegoke, *et al.*, 2020). These vulnerabilities can compromise navigation accuracy and, by extension, the safe operation of AVs, necessitating robust measures to secure these critical systems.

GNSS signals are inherently weak—often likened to the faint glow of a 25-watt light bulb seen from a distance of approximately 20,000 kilometers. This inherent weakness renders the signals exceptionally vulnerable to a multitude of unintentional interferences. For example, physical obstructions such as walls and ceilings in enclosed spaces like garages and tunnels can disrupt signal reception. In urban environments, high-rise structures give rise to multipath challenges; signals reflecting off buildings result in path length variations that introduce significant errors into the positioning data (Spilker, 1996; Spilker Jr *et al.*, 1996). Atmospheric phenomena further compromise GNSS reliability. Ionospheric and tropospheric effects—including scintillation, solar activity, and delays caused by variations in temperature, humidity, and air density—result in excess path delays and signal refraction (Hofmann-Wellenhof, Lichtenegger and Collins, 2012; Baldysz *et al.*, 2023; Kumar *et al.*, 2023). Additionally, segment errors, such as those arising from incorrect data uploads or faults in space vehicles, further contribute to inaccuracies, thereby complicating the provision of robust Positioning, Navigation, and Timing (PNT) services (Montenbruck, Steigenberger and Hauschild, 2020).

In parallel with these unintentional vulnerabilities, GNSS faces severe risks from intentional threats. Spoofing attacks—where counterfeit GNSS signals are broadcast to mimic legitimate ones—can deceive receivers into computing erroneous positions, velocities, and timing data (Dasgupta *et al.*, 2022a; Altaweel, Mukkath and Kamel, 2023; Crosara *et al.*, 2024). Jamming attacks, which inundate the GNSS frequency bands with high-power radio emissions, can completely deny access to authentic signals (Ding, Chen and Ding, 2023). These deliberate forms of interference pose significant risks, especially when compounded by the already fragile nature of GNSS signals. Even alternative navigation aids, such as high-definition map-based systems, are not immune to exploitation by malicious actors who can manipulate an AV's route planning (Elghazaly *et al.*, 2023).

Recent research has categorized GNSS vulnerabilities into four primary types: atmospheric effects, multipath errors, segment errors, and interference (both intentional and unintentional). Atmospheric effects arise as signals pass through the ionosphere and troposphere, where they are delayed and refracted—challenges that are mitigated using empirical models such as the Saastamoinen model (J Saastamoinen, 1972; J. Saastamoinen, 1972) and advanced mapping functions including GPT2, GPT3, and VMF1 (Boehm, Werl and Schuh, 2006; Lagler *et al.*, 2013; Landskron and Böhm, 2018). Multipath errors, caused by signal reflections off nearby structures, are addressed through improved antenna designs and adaptive filtering techniques (Liu *et al.*, 2023; Xin, Geng and Hsu, 2024). Segment errors, originating from the space, user, and control segments of the GNSS, necessitate continuous enhancements in receiver design, differential corrections, and augmentation systems ('Springer Handbook of Global Navigation Satellite Systems', 2017). Finally, interference—whether unintentional (e.g., reamplified signals from UHF/VHF

sources) or intentional (e.g., spoofing and jamming)—requires integrated defenses that combine cryptographic authentication, anomaly detection, machine learning algorithms, and redundant sensor systems (*Two years since the Tesla GPS hack - GPS World : GPS World*, no date; *Using Inertial Systems to Overcome GPS Spoofing - KVH Mobile World*, no date; Qiao *et al.*, 2023).

In recognition of these escalating challenges, the U.S. Federal Government has taken decisive measures to secure PNT services. The Presidential Order issued on February 18, 2020, explicitly called for "Strengthening National Resilience Through Responsible Use of Positioning, Navigation, and Timing Services (*Federal Register :: Strengthening National Resilience Through Responsible Use of Positioning, Navigation, and Timing Services*, no date)." This directive has catalyzed efforts across federal agencies—including the U.S. Department of Homeland Security and the Department of Transportation—to bolster the resilience of GNSS and related PNT infrastructures (*PNT Program | Homeland Security*, no date; U.S. Department of Homeland Security, 2021).

This report presents the outcomes of an extensive investigation into GNSS vulnerabilities and the development of robust cyber-resilient solutions for AV navigation. The research encompassed a detailed analysis of GNSS receiver weaknesses and the design of intelligent slow-drifting spoofing attacks that subtly manipulate navigation data to misdirect vehicle positioning (Dasgupta *et al.*, 2024). In response to these threats, advanced multi-sensor fusion techniques were implemented to integrate data from onboard sensors—such as accelerometers, speedometers, and gyroscope sensors—with GNSS outputs (Dasgupta, Shakib and Rahman, 2024). Deep learning (DL) methodologies, particularly Long Short-Term Memory (LSTM) neural networks, were employed to accurately predict vehicle motion and detect anomalies indicative of spoofing attacks. Moreover, a resilient alternative navigation framework was developed using Geographic Information Systems (GIS) and landmark-based navigation approaches (Dasgupta, 2024). This framework leverages 2D maps, transportation landmarks, and detailed road network topology to provide accurate localization in GNSS-denied environments.

The overarching objective of this project is to fortify the cybersecurity and operational integrity of autonomous vehicle navigation systems by integrating sophisticated attack modeling, sensor fusion-based detection using advanced DL techniques, and resilient alternative navigation strategies. The findings reported herein contribute to both a deeper theoretical understanding of GNSS vulnerabilities and the practical development of cyber-resilient solutions that align with federal mandates to strengthen PNT resilience and secure critical transportation infrastructure.

This report is organized into five chapters. Chapter 1 introduces the research context, outlines the problem statement, and defines the project objectives. Chapter 2 provides an extensive literature review, critically examining current research on GNSS vulnerabilities, spoofing techniques, sensor fusion methodologies, DL applications, and alternative navigation strategies. Chapter 3 details the methodologies and experimental setups used to design and implement the intelligent slow-drifting spoofing attacks, the sensor fusion-based detection system employing LSTM neural networks, and the GIS and landmark-based resilient navigation module. Chapter 4 presents the experimental results and analyses, evaluating the performance and efficacy of the proposed solutions under a range of test conditions. Finally, Chapter 5 offers conclusions drawn from the research, discusses the implications for future autonomous vehicle navigation systems, and outlines recommendations for further advancements in cyber-resilient navigation.

CHAPTER 2

Literature Review

This literature review is organized into three main sub-sections that correspond directly to the project objectives: (1) GNSS Spoofing Attack Mechanisms and Slow-Drifting Techniques, which examines the theoretical foundations and practical implementations of subtle spoofing techniques; (2) GNSS Spoofing Detection Techniques, which reviews state-of-the-art methods and emerging trends in detecting deceptive GNSS signals; and (3) Navigation Strategies for GNSS-Denied Environments, which explores alternative localization strategies for maintaining AV navigation when GNSS signals are compromised.

2.1. GNSS Spoofing Attack Mechanisms and Slow-Drifting Techniques

GNSS spoofing attacks can be broadly categorized into two types: Asynchronous attacks and Synchronous attacks (Van Der Merwe *et al.*, 2018). Asynchronous attacks involve producing a spoofed signal with a different timestamp compared to the legitimate signal. One common example of an Asynchronous attack is a meaconing or replay attack, where a legitimate signal from a previous timestamp is retransmitted at a later time to enhance its authenticity (Lenhart, Spanghero and Papadimitratos, 2021). On the other hand, synchronous spoofing attacks aim to generate spoofed signals with timestamps that match those of legitimate signals. These attacks require prior knowledge of the receiver's location information, making them more challenging to execute due to the difficulty in obtaining such critical details. Advanced synchronous attacks, like the nulling attack, go a step further by entirely canceling out authentic signals, creating a clear path for the uncontested transmission of spoofed signals to the target (Psiaki and Humphreys, 2016). An example of a specific type of synchronous attack is the slow-poisoning distance-decreasing attack, which falls under the category of replay attacks. This attack involves altering the transmission time of a signal to modify the pseudorange, thereby deceiving the receiver about the actual distance. In this chapter, we explore relevant literature concerning distance-decreasing attacks and the various detection mechanisms employed to counter such deliberate threats.

Distance-decreasing attacks, like replay attacks, are relatively agnostic to cryptographic advancements in GNSS anti-spoofing (Lenhart, Spanghero and Papadimitratos, 2021). They rely on reducing the target's pseudorange, or perceived distance from each Global Positioning System (GPS) satellite, in order to alter the final calculated position by the target’s receiver. Motallebighomi et al. (Motallebighomi *et al.*, 2023) demonstrate the resistance of distance-decreasing attacks to cryptography by showing how targets can be successfully spoofed up to 4000km away from their true destination (Lenhart, Spanghero and Papadimitratos, 2021). In the study, the target was able to be spoofed within any location such that there was a set of overlapping satellites. This was done by altering the transmission time of the signals, consequently changing the perceived pseudorange of the satellite by the receiver. By changing the transmission time, the attacker need not generate their own navigation messages, thus avoiding detection by cryptographic mechanisms in place. In another study, Zhang et al. (Zhang *et al.*, 2022) demonstrate that a 1-millisecond change in the transmission time introduces a shift in the target receiver’s perceived location of 71.89km (Zhang and Papadimitratos, 2019). The study shows that changes in transmission time of the order of 1-millisecond can change the pseudorange by around 300km in an unprotected L1 C/A GPS signal. The L1 C/A GPS Signal is compatible with all GNSS devices and is optimal for consumers (‘GPS L1 C/A Receiver Processing’, 2022). The aforementioned studies all implemented distance-decreasing attacks that were successfully able to get past cryptographic security protocols. However, there has been no study that implements such an attack on a smaller magnitude that shifts the target’s location slowly from its initial position.

Spoofing threats can also be classified into three distinct types: simplistic, intermediate, and sophisticated (Humphreys *et al.*, 2008). Simplistic attacks involve the use of a commercial GPS signal simulator, coupled with a power amplifier and antenna (Warner and Johnston, 2002). These attacks typically fall under the asynchronous category and resemble signal jamming, forcing the GPS receiver to lose lock and undergo partial or complete reacquisition. Such attacks are relatively easy to detect. Intermediate attacks, on the other hand, employ portable receiver spoofers. The advent of software-defined radio (SDR) technology (Some and Gasiewski, 2023) has made it and straightforward to develop these portable spoofers, giving rise to intermediate-level attacks. In this scenario, the attacker closely tracks the victim AV to gain knowledge about its GPS receiver antenna's position and velocity, allowing precise positioning of the spoofed signals in relation to the legitimate signals at the AV's antenna. Since these attacks are synchronous in nature, they are considerably more challenging to detect. Sophisticated attacks represent the highest level of sophistication, utilizing multiple phase-locked portable receiver-spoofers. These attacks are designed to deceive angle-of-arrival based defensive systems, making them highly formidable and difficult to counter. Various methods can be employed to execute attacks, but their ultimate goal is to manipulate the pseudorange calculation, leading to erroneous location calculations by the AV's GPS receiver.

2.2. GNSS Spoofing Detection Techniques

The techniques of existing GNSS spoofing attack detection methods can be classified into four categories: (i) encryption mechanisms; (ii) codeless-cross-correlation measures; (iii) signal statistics analyses; and (iv) antenna-based methods (Zidan, E. I. Adegoke, *et al.*, 2020). The most common GNSS anti-spoofing methods use encryption algorithms to secure the GNSS signals. Although the military commonly uses the encryption mechanism approach to secure GNSS receivers, this is not a solution due to its high infrastructural, computational, and management cost. The codeless-cross-correlation measures use the correlation among unknown encrypted GPS L1 P(Y) code from multiple receivers to detect a spoofing attack (O'Hanlon *et al.*, 2010, 2013). Note that L1 is the primary GPS carrier signal, and P(Y) code is the precision code. The effectiveness of such a method also depends on the cost of new instruments and associated signal processing complexity when the number of cross-checking GNSS receivers is increased. The GNSS signal statistics analysis-based approaches use different signal features, such as received signal strength (RSS) (Yang *et al.*, 2013), spatial coherency (Daneshmand *et al.*, 2012), pseudo-range measurements, time of advent, and signal parameters estimation to detect GNSS spoofing attacks. Antenna-based methods include detecting a spoofing attack by using multi-antenna GNSS, reduced inertial sensor system (RISS), and Inertial Navigation System (INS) integration (Vagle, Broumandan and Lachapelle, 2017) to perform beat carrier-phase measurement processing using two antennas (Psiaki *et al.*, 2014). A single antenna combined with RISS can also be used to detect spoofing attacks (Hu *et al.*, 2018). These approaches require computationally expensive GNSS signal processing algorithms and sophisticated antenna arrays to ensure high spoofing attack detection accuracy (Zidan, E. I. Adegoke, *et al.*, 2020; Liu *et al.*, 2021).

Besides these approaches, GNSS spoofing attacks can also be detected by comparing vehicle acceleration from Inertial Measurement Unit (IMU) with the GNSS derived acceleration according to (Neish *et al.*, 2018). Although this approach performs well for an aircraft, it is not suitable for surface vehicles due to the low vehicle dynamics signature. In (Manickam and O'Keefe, 2016), the location information derived from IMU sensors (i.e., accelerometer and gyroscope) is compared with the GNSS-derived location for spoofing attack detection. Furthermore, INS has also been used to monitor the position of a vehicle for detecting GNSS spoofing attacks (Manickam and O'Keefe, 2016; Tanıl *et al.*, 2018). INS devices use gyroscope and accelerometer data and calculate the position, orientation and speed of a vehicle using dead reckoning without any input from GNSS. However, INS derived location is less accurate as the measurements from inertial sensors accumulate bias, scale factor, and non-orthogonality errors over time. In addition, multiple antennas are used to identify spoofing attacks through cross-checking GNSS signals (Psiaki *et al.*, 2014).

In addition to the above-mentioned approaches, the development of machine learning (ML) and DL algorithms has recently increased for spoofing attack detection. In (Tanil *et al.*, 2016) a Multi-Layer Perceptron (MLP), a Complex Convolution Neural Networks (CNN), and a simple CNN are used to detect spoofed GNSS signals that demonstrate the potency of using deep neural networks for spoofed signal detection. In (Panice *et al.*, 2017) the authors provide a decision fusion with the K-out-of-N decision rule-based method along with wavelet transformation coefficients. In(Borhani-Darian et al., n.d.)(Borhani-Darian *et al.*, 2020) a Support Vector Machine (SVM) has been used for state estimation and detection of an attack on unmanned aerial vehicles. The early-late phase, delta, and signal level are used as features, together with the K-Nearest Neighbor (KNN) and naïve Bayesian classifier to detect spoofing attacks (Sun *et al.*, 2017). However, these ML and DL algorithms are used to detect an attack in the signal level, and no research has been conducted to predict the distance a vehicle can travel within a timeframe and detect an attack based on that. Only INS-based spoofing attack detection approaches, where an INS-based vehicle's position was compared with the GNSS-based position, are closely related to our study. However, position information from INS sensors is not reliable due to the error propagation of inertial sensors over time. Thus, none of the existing approaches used the location domain information to detect the spoofing attack.

Several existing companies (Levy and Stern, 2019; Broumadan, Kennedy and Schleppe, 2020; Buesnel, 2020; Dries, Pratt and Johnson, 2021; Septentrio, 2024; Trimble, 2024) offer commercial GNSS jamming and spoofing detection services. These companies utilize proprietary algorithms to detect anomalies in the received GNSS signal. These attack detection technologies either use costly local precise atomic clocks or the results of analysis of GNSS signal characteristics for spoofing attack detection.

2.3. Navigation Strategies for GNSS-Denied Environments

In the dynamic and rapidly advancing domain of AV navigation, a diverse array of alternative localization techniques is being explored to enhance AV navigation capabilities. These methodologies can be categorized into seven distinct types (See **Figure 1**): (i) INS, which utilize accelerometers and gyroscopes to determine vehicle position and orientation; (ii) Signal of Opportunity (SOP)-based navigation, offering enhanced signal strength and positioning accuracy from available signals; (iii) Vision-based navigation, relying on cameras and image processing to interpret the vehicle's surroundings; (iv) SLAM -based navigation; (v) Map-based navigation, employing pre-existing maps or physical road markers for guidance; (vi) Landmark-based navigation and (vii) RFID-based navigation, using radio frequency identification for location tracking. These diverse approaches reflect the ongoing innovation and adaptation in the field of AV navigation. The strengths and limitations of these methodologies are presented in **Table 1**.

INS, leveraging IMU, is integral in dead reckoning navigation for ground AVs, especially in environments where GNSS is unreliable or unavailable. By combining data from accelerometers and gyroscopes, INS calculates a vehicle's position and orientation from a known starting point. Despite its independence from external signals, INS is prone to errors over time, a phenomenon known as drift. This has led to innovative error mitigation research, as Seen in (Yang *et al.*, 2021; Liu *et al.*, 2024; Yu *et al.*, 2024), who explore NLOS error compensation and integrated GNSS/IMU/Vision systems, respectively. However, despite these advancements, INS is still not perfect. The primary challenge with INS is the inherent error accumulation or drift over time, caused by the minute inaccuracies in the measurements of accelerometers and gyroscopes within the IMUs. This drift can lead to significant deviations from the actual path, particularly in prolonged operations or environments with limited external navigational aids. Therefore, while INS systems offer a high degree of autonomy and reliability in challenging environments, their susceptibility to error accumulation underscores the ongoing need for continuous innovation and improvement in navigational technology.

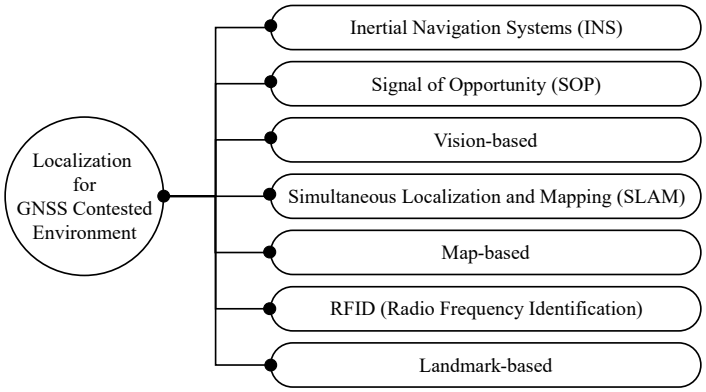


Figure 1: Localization for GNSS contested environment.

SOP-based localization is another approach that utilizes existing communication and broadcasting signals for positioning and navigation. This method capitalizes on the widespread availability of various signals, including Low Earth Orbit (LEO) satellites, cellular networks (3G/4G/5G), WiFi, and Bluetooth. LEO satellites, orbiting closer to Earth, provide stronger and more accurate signals for localization compared to traditional geostationary satellites, but they can be limited by their smaller coverage area and the need for a larger constellation for continuous global coverage (Yang *et al.*, 2021; Khalife and Kassas, 2023; Saroufim, Hayek and Kassas, 2023; Zhao *et al.*, 2023; Kassas, Khairallah and Kozhaya, 2024). Cellular networks, through 3G/4G/5G technologies, offer widespread coverage and are particularly useful in urban areas. However, their localization accuracy can be affected by signal multipath and urban canyon effects (Abdallah, 2023; Abdallah, Khalife and Kassas, 2023; Kassas and Abdallah, 2023; Saroufim, Hayek and Kassas, 2023). WiFi-based localization leverages the ubiquity of WiFi networks, using signal strength and triangulation for positioning. While convenient, its accuracy is often limited by the density and range of WiFi access points. Similarly, Bluetooth, especially with the advent of Bluetooth Low Energy (BLE), is used for micro-location and indoor navigation. Its primary limitation is the short range and susceptibility to interference. Each of these SOP sources has unique advantages in terms of availability and ease of integration, but they also face challenges such as signal interference, varying degrees of accuracy, and dependency on the existing infrastructure.

Table 1: Localization techniques for GNSS denied environments.

Method	Strengths	Limitations
INS	Independent of external signals, calculates position and orientation from a known start point.	Prone to error accumulation or drift over time, leading to inaccuracies.
SOP	Utilizes a variety of signals, enhanced signal strength and accuracy.	Affected by signal interference, variable accuracy, and dependency on existing infrastructure.
Vision-based	Interprets complex environments through cameras, advanced in pose estimation.	Requires high computational resources, limited by camera's field of view and environmental conditions.
SLAM	Maps unfamiliar environments while localizing the vehicle within it.	Relies heavily on the quality and consistency of sensor data, challenged by dynamic environments.
Map -based	Offers high accuracy using HD maps and physical road markers.	High data and computational demands, challenges in dynamic environments.
RFID-based	Effective in environments like tunnels or mines, enhances localization.	Dependent on tag density and placement, susceptible to signal interference.
Landmark-based	Utilizes identifiable features for localization, effective in urban areas	Limited by the ubiquity and consistency of landmarks, challenges in dense or featureless areas

Vision-based navigation is a rapidly evolving component in the field of autonomous systems, particularly useful in environments where traditional navigation methods like GNSS fall short. This technology primarily utilizes visual data captured from cameras to navigate and interpret complex environments.

Significant strides have been made in enhancing the reliability and accuracy of these systems, with notable developments in pose and reliability estimation using advanced machine learning techniques like convolutional neural networks and Rao–Blackwellized particle filters (Naoki Akai and Murase, 2018). In addition, research on embodied navigation agents has provided insights into the interpretative and interactive capabilities of autonomous systems, contributing to a deeper understanding of their operation in various contexts (Anderson *et al.*, 2018) . The design of control systems, especially PID (Proportional Integral Derivative) control systems, is another crucial aspect that ensures the stability and precision of vision-based navigation systems (Ang, Chong and Li, 2005). Innovations, such as the introduction of NetVLAD, a CNN (Convolutional Neural Network) architecture for place recognition, have marked significant advancements, enabling efficient and accurate place recognition, a critical component for effective navigation (Arandjelović *et al.*, 2018). Challenges in accurately determining camera positions for visual camera re-localization have also been a focus, highlighting the limits of pseudo ground truth and its implications for practical applications (Brachmann *et al.*, 2021). Finally, the application of vision-based navigation in automotive settings, particularly for accurate visual localization, illustrates its transformative potential, especially in enhancing the capabilities of self-driving cars. These developments collectively underscore the growing importance and sophistication of vision-based localization in the realm of autonomous navigation (Brosh *et al.*, 2021).

Simultaneous Localization and Mapping (SLAM) (Yousif, Bab-Hadiashar and Hoseinnezhad, 2015; Hess *et al.*, 2016; Taketomi, Uchiyama and Ikeda, 2017; Milz *et al.*, 2018; Ramesh *et al.*, 2021) can play a pivotal role in the navigation of autonomous ground vehicles, especially in environments where GNSS is unreliable or unavailable. It enables these vehicles to map unfamiliar environments while simultaneously determining their own position within the map. However, SLAM technology has limitations that impact its effectiveness in autonomous ground vehicles. One significant challenge is the reliance on the quality and consistency of sensor data. Inaccuracies in sensor readings, such as from LiDAR or cameras, can lead to errors in map construction and vehicle localization. Additionally, dynamic environments pose a challenge, as SLAM systems must continuously update the map to account for changes, like moving obstacles or varying terrain conditions.

Map-based navigation, particularly utilizing High-Definition (HD) maps (Dooley, 2018; Geomate, 2022; Křehl\’\ik, Vanžura and Skokan, 2023; Odukha, 2023), has become increasingly relevant in the field of AV navigation. HD maps contain detailed environmental information, necessitating high data volume and computational effort for analyzing and fusing map data with sensor information. Despite their complexity and the challenges in evaluating performance, especially in urban areas, HD maps are favored for their accuracy and availability. LiDAR sensors play a crucial role in creating HD maps (Aldibaja, Suganuma and Yoneda, 2017). For instance, LiDAR point cloud data, combined with post-processed localization measurements, can be assembled to create comprehensive HD maps suitable for localization through map matching. Several studies have focused on accumulating multiple LiDARs to generate point clouds for high-definition 3D maps with georeferencing coordinate information through GNSS/IMU systems (Ilci and Toth, 2020). These maps aid in performing LiDAR-based localization, although they may not always cover the localization process in depth (Tao *et al.*, 2022). To achieve robust localization, especially in complex urban areas with semi-static and dynamic objects, it's essential to ensure consistency and manage discrepancies between map and sensor data. Techniques like Fourier-Mellin transformation have been proposed for map matching to estimate the vehicle's global pose, combined with tests like the cumulative sum test for checking map consistency (Rohde *et al.*, 2016; Li *et al.*, 2020). Sensor fusion approaches, incorporating LiDAR odometry, wheel odometry, map matching results, and GNSS, have also been utilized to enhance localization accuracy (Li *et al.*, 2020). However, map-based localization faces challenges due to changes in maps during different times of the day or in harsh weather conditions. Features within the map may appear differently, necessitating the use of sensors less prone to weather errors or techniques for

reconstructing the map in real-time. Approaches like principal component analysis to reconstruct LiDAR images and enhance map quality have been explored to improve localization accuracy in adverse conditions. In landmark-based navigation, distinct environmental features like landmarks or road marks as robust references for localization are utilized, particularly effective in urban areas where dynamic objects such as vehicles and pedestrians make generic feature use challenging. This method stands out for not requiring a detailed HD map, instead relying on sparse maps that contain positions of landmarks or road marks. Landmarks such as trees, traffic light poles, street light poles, and traffic signs have been employed for vehicle localization (Sefati *et al.*, 2017; Wang *et al.*, 2018; Weng *et al.*, 2018; Schaefer *et al.*, 2019), offering strong accuracy, especially in small urban areas with limited GNSS reliability. For instance, the utilization of pole-like landmarks and traffic light information as position markers has demonstrated localization accuracy with errors less than 30 cm for LiDAR and below 50 cm for stereo cameras (Sefati *et al.*, 2017). These methods, while effective, face the limitation of landmark non-ubiquity, as not all regions possess easily identifiable or consistent landmarks. Additionally, for lateral localization lane markings are used. Notable approaches include using HD maps and monocular vision cameras to detect the distance between a vehicle and lane markings, integrating this data with GNSS coordinates via the Kalman Filter (KF) for enhanced localization accuracy. Additionally, pioneering methods have employed front-looking cameras to detect lane markings, updating particle weights for the Particle Filter (PF) algorithm, and integrating HD maps, and vehicle odometry to accurately ascertain the vehicle's position, demonstrating the effectiveness of combining various sensing modalities and map-matching techniques for robust vehicle localization (Shin *et al.*, 2014; Bauer, Alkhorshid and Wanielik, 2016; Tao *et al.*, 2022). Road marking such as crosswalks, stop lines have also been used for localization to aid the navigation of the AV (Hata and Wolf, 2014, 2015). In environments like dark tunnels or mines, where conventional navigation methods are less effective, RFID technology has been employed to enhance localization and mapping. By strategically placing RFID tags, researchers have integrated the data with robot odometry and laser scans for improved navigation. This approach has also been applied to multi-robot exploration and path planning (Kleiner, Prediger and Nebel, 2006; Ziparo *et al.*, 2007; Vorst *et al.*, 2008; Beinhofer, Kretzschmar and Burgard, 2013). However, the effectiveness of RFID-based navigation is contingent on the density and placement of the tags, which might not be uniformly feasible across all environments. Furthermore, RFID systems can be susceptible to signal interference and may not provide the level of detail or accuracy achievable with more advanced sensing technologies, posing limitations in complex spatial mapping and real-time navigation adjustments.

While existing methods for AV localization and navigation exhibit proficiency in specific scenarios, they often face constraints in terms of robustness, complexity, and computational load.

CHAPTER 3

Methods

Methods section is divided into three sections that align with the project goals. The first section, Design and Implementation of Slow-Drift GNSS Spoofing Attacks for Autonomous Vehicles, explains how we create subtle spoofing attacks that slowly change a vehicle's position without being detected right away. The second section, Experimental Validation of a Sensor Fusion-Based GNSS Spoofing Attack Detection Framework, describes how we combine data from multiple in-vehicle sensors, such as accelerometers and gyroscopes, to detect any anomalies caused by spoofing attacks, and explains the experiments conducted to test this detection system. The third section, Development of a GIS and Landmark-Based Navigation System for Autonomous Vehicles in GNSS-denied Urban Environments, outlines how we built a backup navigation system that uses maps and fixed landmarks to help vehicles navigate accurately in urban areas where GNSS signals are unreliable. Together, these methods offer a complete approach to creating, detecting, and reducing the risk of GNSS spoofing attacks on autonomous vehicles.

3.1. Design and Implementation of Slow-Drift GNSS Spoofing Attacks for Autonomous Vehicles

3.1.1. Receiver Location Calculation

Emulating the spoofed path necessitates the development of a software program capable of utilizing satellite signal data to calculate the receiver's location. This software program will play a pivotal role in determining the desired pseudorange values required to emulate the spoofed location. Subsequently, the simulated spoofed signals can be generated based on these calculated pseudorange values. By effectively implementing this process, the spoofed path can be generated to deceive the GPS receiver and manipulate the navigation system. Note that only GPS satellite signals and corresponding data are used for receiver location calculation in this study.

The pseudocode of the GPS receiver location calculation is presented below. The calculation is done using three steps. In the first step, the position of the satellite is calculated. Then the distance of the receiver from the satellite is calculated and finally receiver location and the receiver clock bias are calculated. GPS satellite signal data are acquired from the Receiver Independent Exchange Format (RINEX) (Gurtner, 2018) observation and navigation files. The navigation file contains the clock correction coefficients, ephemeris, and integrity (CEI) data as shown in **Table 2**. The observation file contains the pseudorange and carrier phase data. Pseudorange is the calculated distance between the receiver and the satellite based on the time it takes the signal to reach the receiver from the satellite and multiplying it by the speed of light (See **Equation 1**).

$$p_r^s(t_A) = c\,(t_A - t_E) \tag{1}$$

Where, $p_r^s(t_A)$ is the pseudorange or distance between satellite(s) and receiver r at time t_A, which stands for the time of arrival. c is the speed of light and t_E is the time when the signal was emitted from the satellite.

As the receiver clock is not as accurate as the satellite atomic clock which introduces errors in the distance calculation, the range is called pseudorange. The receiver used correlations of the satellite PRN codes for the pseudorange calculation. The location of the GPS satellite is calculated using the ephemeris data in the navigation file and using the equations provided in Table 7.9 of the reference ('Springer Handbook of Global Navigation Satellite Systems', 2017). The values of constants used for the calculations are presented in **Table 3**. The location of the satellite is first determined in Earth-centered Earth-fixed (ECEF) coordinate system and then converted to the World Geodetic System (WGS) 84. Satellite or space vehicle (SV) clock bias is also corrected using the SVclockBias, SVclockDrift, and SVclockDriftRate data. To accurately calculate the distance of the satellite, the receiver Ionospheric and Tropospheric corrections are also

performed. Least square regression is used to solve for the receiver location from satellite positions. At least data from four satellite signals data are required to solve for four unknowns i.e., user latitude, longitude, altitude, and the receiver clock bias.

Algorithm 2.1 Receiver Location Calculation

```
xu ← [0, 0, 0]                                   ▷ initial estimate for user position
b ← 0                                            ▷ initial estimate for user clock bias
SV s ← all observed satellites
for sat in SVs do
    dsv ← satellite clock bias(sat)
    C ← Speed of Light
    sat, pseudorange ← sat.pseudorange + C ∗ dsv − C ∗ sat.T gd
end for
Xs ← [ ]                                         ▷ satellite position matrix
Pr ← [ ]                                         ▷ pseudorange matrix
while norm(dx) > 0.1 and norm(db) > 1 do
    for sat in SVs do
        cpr ← sat.pseudorange − b                ▷ corrected pseudorange
        xs ← satellite position(sat)
        Xs append xs
        Xs append xs
    Xs append xs end for
    x , b ← estimate position(Xs, Pr, length(SV s), xu, b)
    dx ← x_ − xu
    db ← b_ − b
    xu ← x_
    b ← b_
end while
return xu, b
```

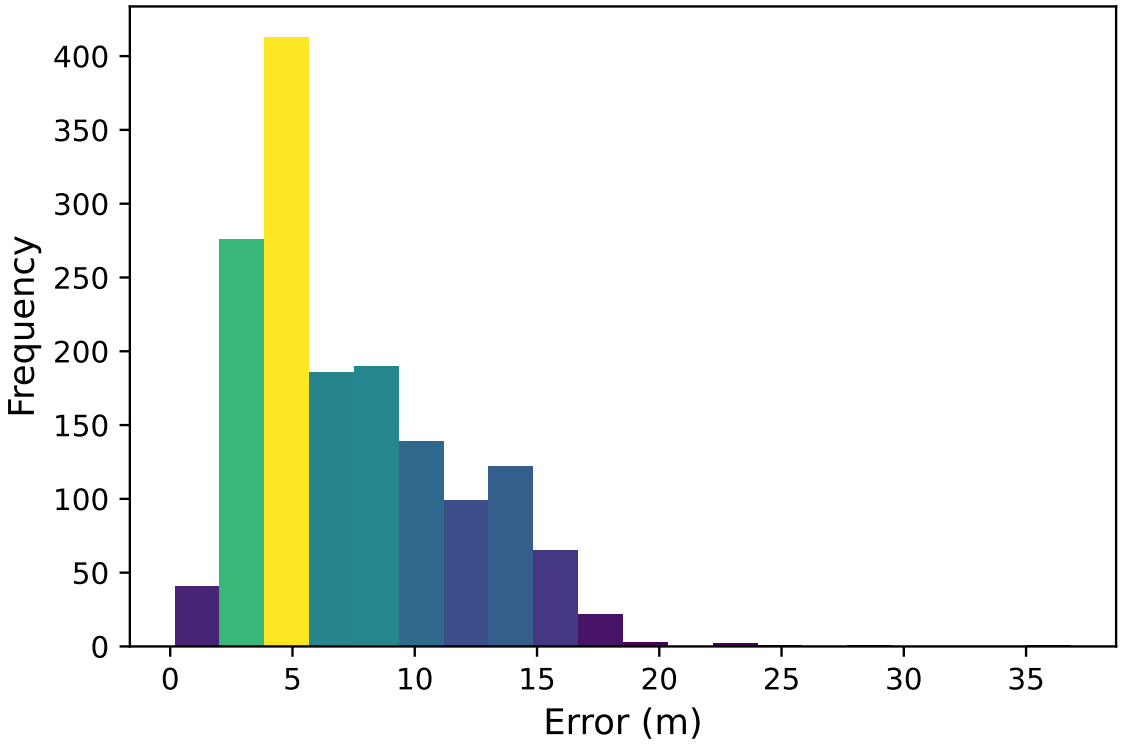


Figure 2: Location calculation error distribution.

Table 2: CEI variables from RINEX navigation file.

Measurements	Description	Sample Data
Satellite ID	Satellite system (G) number (PRN)	G18
Time	GPS Clock Time	7/14/23 22:00
SVclockBias	Space Vehicle clock bias correction coefficient	-3.46E-04
SVclockDrift	Space Vehicle clock drift correction coefficient	-1.42E-11
SVclockDriftRate	Space Vehicle clock drift rate correction coefficient index	0.00E+00
IODE	Issue number of the satellite ephemeris data set, Issue of Data, Ephemeris	1.46E+02
Crs	Amplitude of the Sine Correction Term to the Orbit Radius	-5.00E+01
DeltaN	Mean Motion Difference from Computed Value at Reference Time	4.37E-09
M0	Mean Anomaly at Reference Time	-3.71E-01
Cuc	Amplitude of Cosine Harmonic Correction Term to the Argument of Latitude	-2.54E-06
Eccentricity	Eccentricity	3.45E-03
Cus	Amplitude of Sine Harmonic Correction Term to the Argument of Latitude	1.20E-06
sqrtA	Square root of the semimajor axis	5.15E+03
Toe	Time of Ephemeris	5.11E+05
Cic	Amplitude of the Cosine Harmonic Correction Term to the Angle of Inclination	1.30E-08
Omega0	Longitude of Ascending Node of Orbit Plane at Weekly Epoch	1.52E+00
Cis	Amplitude of the Sine Harmonic Correction Term to the Angle of Inclination	-7.45E-09
i0	Inclination Angle at Reference Time	9.74E-01
Crc	Amplitude of the Cosine Harmonic Correction Term to the Orbit Radius	3.60E+02
omega	Argument of Perigee	3.13E+00
OmegaDOT	Rate of Right Ascension	-8.27E-09
IDOT	Rate of Inclination Angle	5.04E-11
CodesL2	Codes on L2 channel	1.00E+00
GPSWeek	GPS week number	2.27E+03
L2PFlag	L2P data flag	0.00E+00
Svacc	Satellite Vehicle Accuracy	2.00E+00
health	Satellite health	0.00E+00
TGD	Group Delay Differential	-8.38E-09
IODC	Issue number of the satellite clock data set	4.02E+02
TransmissionTime	Transmission time of the message	5.04E+05

Table 3: List of constants and values used for receiver location calculation.

Constant	Details	Value	Unit
c	Speed of light	299792458	m/s
μ	WGS 84 value of the Earth's gravitational constant	3.986005×10^{14}	m^3/s^2
$\ddot{\Omega}_e$	WGS 84 value of the Earth's rotation rate	$7.2921151467 \times 10^{-5}$	rad/s
π		3.1415926535898	
R	Radius of Earth	6372.8	km

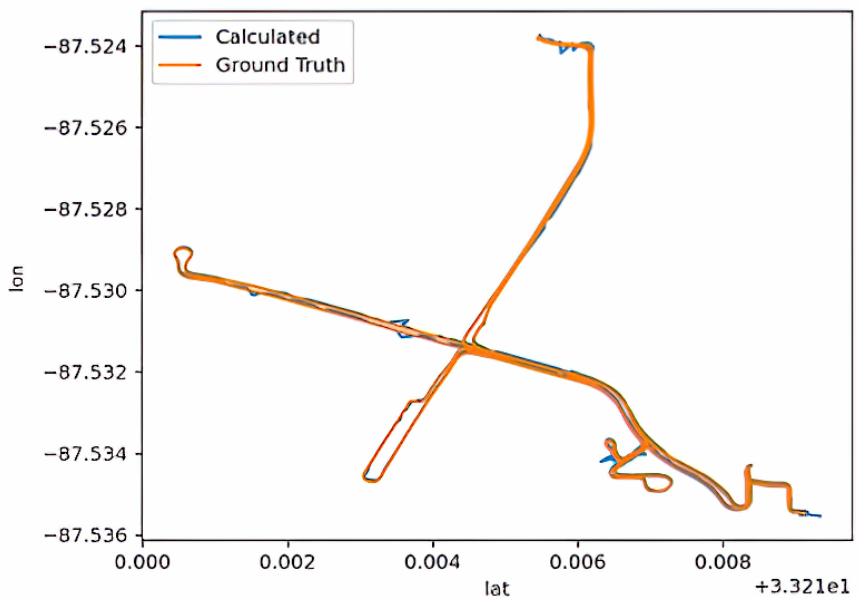


Figure 3: Ground truth and calculated path.

The performance of the receiver location calculation software is checked against the ground truth location. As shown in **Figure 3**, our research team has collected GPS signal data using a NovAtel CPT7700 integrated receiver with TerraStar corrections at the University of Alabama (UA) campus roadway network. This receiver has a localization accuracy of 2.5 cm. Data was stored in the device and then converted to RINEX format. The RINEX files are then used for the receiver location calculation. The distribution of error which is the distance between the location calculated by the presented algorithm and the location solution from the NovAtel receiver is presented in **Figure 2**. The distance or error is calculated using the Haversine formula (See **Equation 2**).

$$d = 2r \sin^{-1}\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\psi_2 - \psi_1}{2}\right)}\right) \quad (2)$$

where the distance, d, is described as the shortest distance between 2 points on a sphere represented as geodetic (lat/lon) coordinates, (φ_1, ψ_1) and (φ_2, ψ_1). The radius, r, is the radius of the earth (6372.8 km). The distribution is right-skewed or positively skewed i.e., most of the error values are concentrated in the lower values. The most populated bins have errors of less than 6 m. There are fewer occurrences of error more than 18 m. The minimum and the maximum error are 0.18 m and 36.85 m, respectively. The mean and median values are 7.40 m and 6.17 m respectively. Note that location data are calculated without considering real-time kinematics correction, which is included in built-in NovAtel CPT7700, and lead to higher location error. However, **Figure 3** proves how closely the receiver location calculation algorithm matches with the ground truth path which proves its ability to be used for detecting and generating spoofed routes. The x-axis shows the latitude (lat), and the y-axis shows the longitude (lon). These results show that

the presented algorithm has the capability for estimating the location of the receiver and using it for further emulating the spoofing attacks.

3.1.2 GPS Spoofing Attack Modeling

The schematic representation of the slow drift stealthy attack framework is illustrated in **Figure 5.** The attacker initiates the attack armed with prior knowledge of the spoofed path locations. Throughout the attack, the attacker closely tracks the AV to ensure that both the AV and the attacker can observe the same satellites. The attacker's receiver captures legitimate GPS signals and records the navigation and observation data on the local computer. Navigation data comprises information sourced from the RINEX navigation file, while observation data includes the pseudorange information calculated by the attacker's receiver. To generate the desired spoofed pseudorange values, the true pseudorange data extracted from the observation data is combined with the pre-defined spoofed location and fed into the Spoofed Pseudorange Emulator (SPE). The SPE employs experiment-derived correlations or machine learning models to estimate pseudoranges that would yield the spoofed location. Subsequently, the calculated spoofed pseudoranges, along with legitimate CEI (Clock, Ephemeris, and Integrity) variables from the navigation data, are fed into the GPS receiver location calculation algorithm to determine the spoofed location. The calculated spoofed location is then compared to the desired spoofed location. If they match, the spoofed pseudoranges are transmitted to the Spoofer GPS Signal Simulator. In this phase, the GPS signals are manipulated by altering the signal generation time and carrier phase, ensuring that the AV receiver's localization solution yields the spoofed location. Moreover, the spoofed signal mirrors the satellites visible in the legitimate signal, exhibiting a seamless transition with no abrupt changes in signal or message properties after the attacker locks onto the AV's receiver. The pre-determined frequency of the spoofed location changes is designed to mimic a gradual drift from the AV's actual route. Finally, the simulated GPS signals are transmitted to the AV's GPS receiver, effectively executing the slow drift stealthy attack and detection.

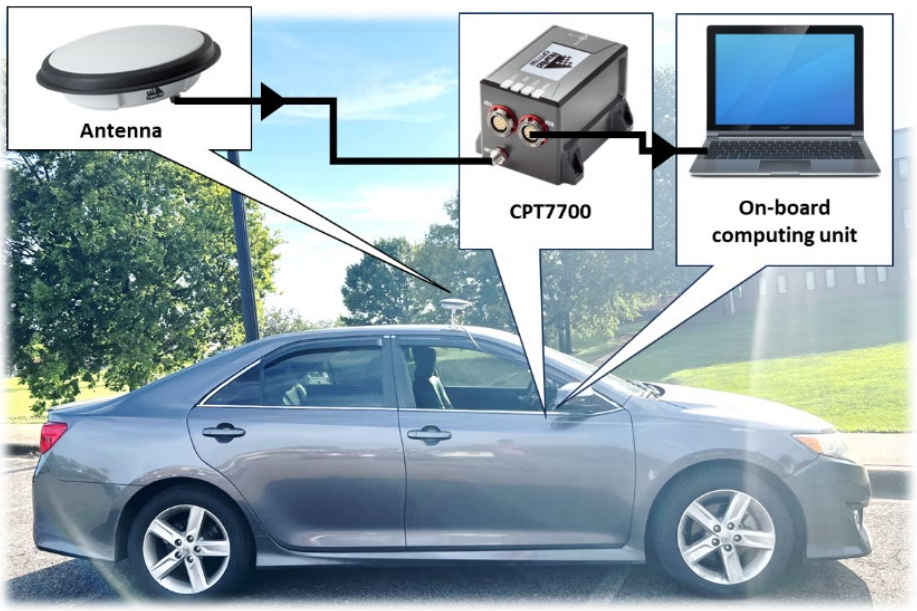


Figure 4: Vehicle with a NovAtel CPT7700 integrated receiver with TerraStar corrections

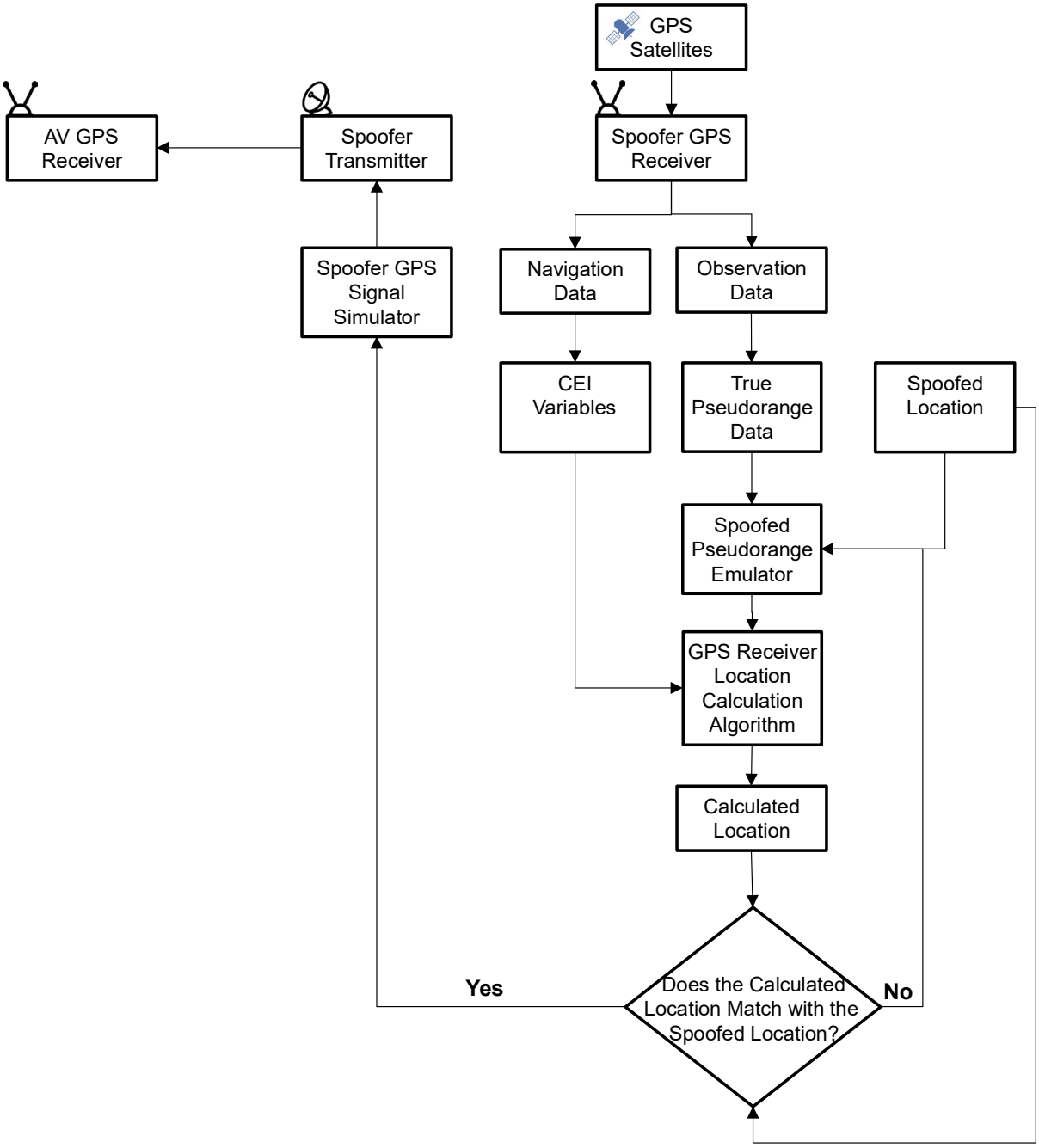


Figure 5: Attack modeling framework.

3.1.3 Experimental Setup

The experimental setup for creating the dataset for creating GPS spoofing attack takes place on the premises of the University of Alabama campus. To simulate real-world conditions, a moving vehicle is used as the target, equipped with a GPS receiver as shown in **Figure 4**. Specifically, a NovAtel CPT7700 integrated receiver with TerraStar corrections is employed for this purpose. This receiver boasts an OEM7700 receiver from Hexagon-NovAtel, complemented by a high-performing Honeywell HG4930 Micro Electromechanical System (MEMS) Inertial Measurement Unit (IMU) and a VEXXIS® GNSS-800 Series Antenna. With the capability to track multi-GNSS signals across all frequencies simultaneously, this receiver is well-suited for the experiments. The localization solution output frequency is set at 1 Hz, and the receiver exhibits a high localization accuracy of 2.5 cm. After conducting the experiments, the data are converted to the RINEX format using the NovAtel application suite. As shown in **Figure 6**, The driving route carefully replicates an urban road structure, aiming to imitate real-world scenarios. To effectively assess the AV's navigational performance, the route incorporates three basic maneuvers, including going straight, taking a left turn, and taking a right turn. Ensuring the integrity of the experimental setup, the visibility of the same set of satellites is maintained throughout the entire duration of the experiment. To capture comprehensive GPS data for the entire route, all relevant information is diligently stored in the local storage system. This robust experimental setup lays the foundation for analyzing the effects of the slow drift stealthy GPS spoofing attack on the AV's navigation performance. By closely monitoring the AV's reaction to the manipulated GPS signals, valuable insights can be gained, enabling the development of effective defense mechanisms against such sophisticated attacks in the future.

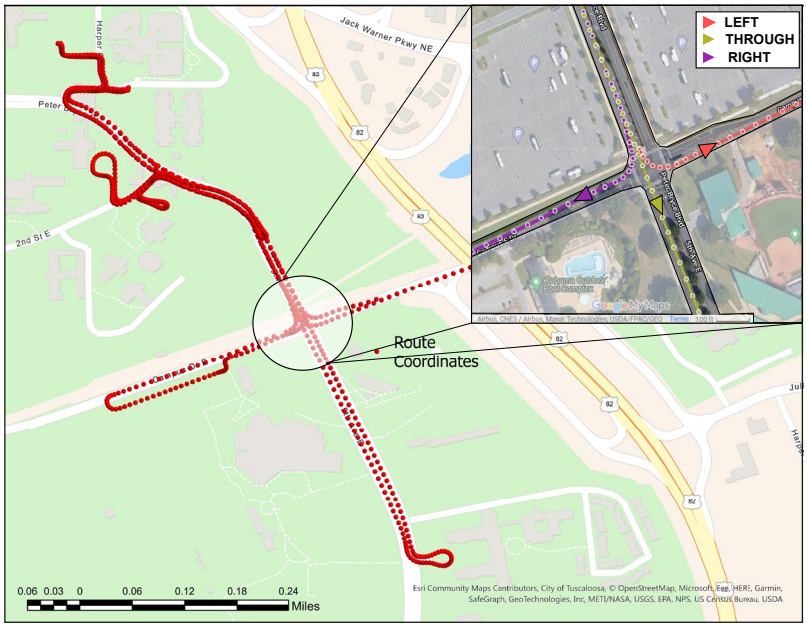


Figure 6: Driving route during the field experiment at the University of Alabama Campus, AL.

3.2 Sensor Fusion-Based GNSS Spoofing Attack Detection Framework

3.2.1 Attack Detection Framework

Figure 7 presents the experimental framework, which employs the GNSS spoofing attack detection framework developed by (Dasgupta *et al.*, 2022b). This framework integrates data from in-vehicle sensors—GNSS, accelerometer, gyroscope, and speedometer—using two concurrent strategies to create a unified and resilient approach for detecting GNSS spoofing attacks. The core of the system is the CPT 7700

Sensor Suit, which includes GNSS, and an IMU featuring an accelerometer and a gyroscope for measuring acceleration and orientation. Additional data such as speed from Synchronized Position Attitude Navigation (SPAN), which provides advanced positioning and navigation capabilities by combining GNSS data with inertial measurements is also integrated. This integration enhances the accuracy and performance of GNSS by compensating for situations where satellite signals are weak or obstructed, such as in urban canyons or under dense foliage, providing continuous, accurate positioning, velocity, and attitude (orientation) information. Speedometer data can be used instead of SPAN data if available. The data collected by these sensors are stored in a MongoDB Database. MongoDB is chosen for its superior performance, scalability, availability, and flexibility compared to SQL databases. (Hevo Data, 2024). The spoof database contains latitude and longitude of the spoofed routes. The detection suite utilizes two strategies for spoofing attack detection. The first strategy develops a vehicle state prediction model by training a LSTM deep recurrent neural network with attack-free data from the SPAN (speed), accelerometer, and gyroscope. It predicts the location shift between consecutive timestamps and monitors the vehicle's motion state using speedometer data. By comparing the predicted location shift with the GNSS-based location shift, the system can detect a spoofing attack. If the absolute difference between perceived and predicted location shifts are greater than the error threshold an alert is triggered for a potential GNSS spoofing attack, indicating that the integrity of the location and navigation data may be compromised. Vehicle motion state is also detected using SPAN speed data and compared with GNSS-based motion state detection. If the GNSS derived motion state doesn't match with the SPAN based motion state an alert is triggered. The second strategy, which focuses on recognizing left and right turns, trains a Random Forest (RF) algorithm with vehicle route data to identify patterns of turning maneuvers. This strategy combines gyroscope and GNSS turning information, to accurately classify turns. If GNSS-based turn detection result doesn't match with the gyroscope-based turn detection result an alarm is triggered. Motion state of the vehicle is also checked in this strategy.

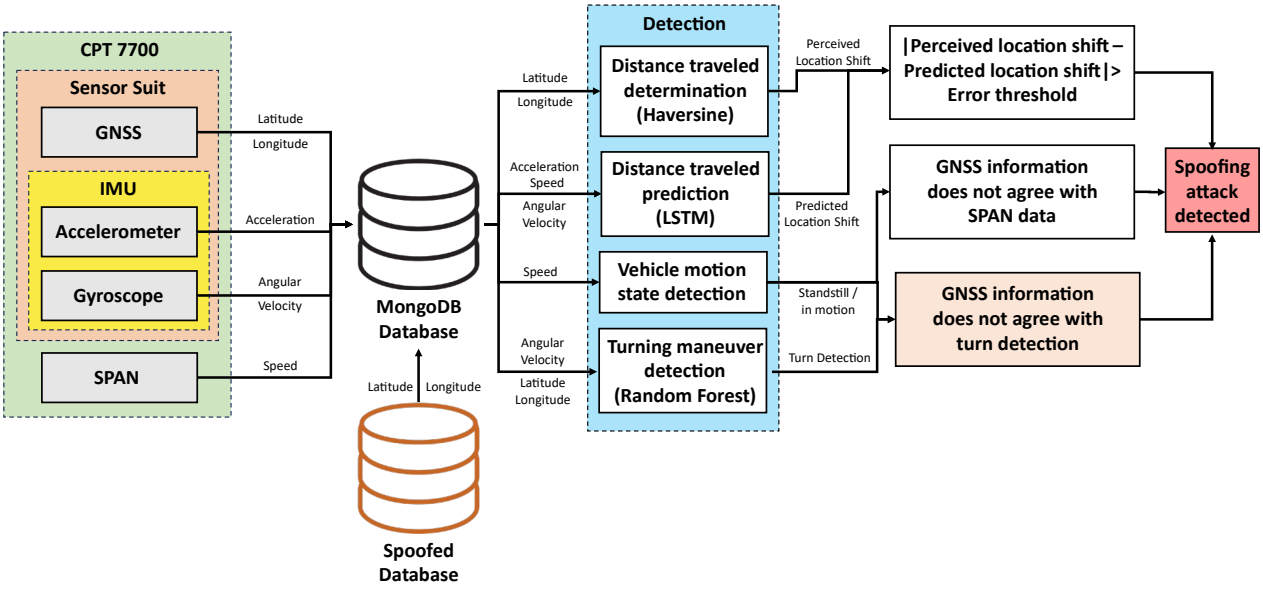


Figure 7: Detection framework for field experiments.

3.2.2 Experimental Setup

Figure 8 presents an experimental set-up to evaluate the sensor fusion-based spoofing attack detection performance using two attack scenarios (turn-by-turn and wrong turn). A Toyota Camry 2014 SE car is equipped with NovAtel CPT7700 with GNSS and Inertial Navigation System (INS) technology. CPT7700 contains an OEM7700 multi-frequency GNSS receiver, which can track current and upcoming GNSS

constellations, including GPS, GLONASS, Galileo, BeiDou, QZSS, and IRNSS. It also includes TerraStar Correction Services with RTK for centimeter-level real-time positioning accuracy. It is equipped with NovAtel's SPAN technology for continuous 3D position, velocity, and attitude. The performance of the GNSS receiver is being presented in **Table 4**. CPT7700 is also equipped with the high-performing Honeywell HG4930 Micro Electromechanical System (MEMS) IMU containing gyroscope and accelerometer. The performance of the IMU is presented in **Table 5**. A CPT7 I/O2 cable is used to connect CPT7700 with a Windows-10 based onboard computing unit with the USB port, which supports a hi-speed (480Mb/s) data rate. The Hexagon GNSS-850 high-precision antenna with superior tracking performance is being used. It has the ability to track low-elevation satellites while maintaining a high gain for higher-elevation satellites, making it suitable for applications where the sky is partially visible, such as operating close to tree lines, under foliage, or in urban canyons.

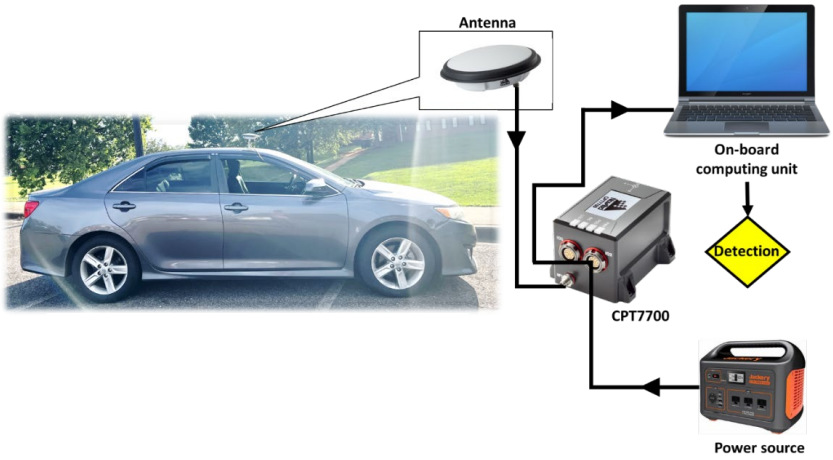


Figure 8: Hardware setup.

Table 4: GNSS receiver performance.

Positioning	Accuracy (RMS)
Single Point L1	1.5 m
Single Point L1/L2	1.2 m
SBAS	60 cm
DGPS (code)	40 cm
TerraStar-C PRO	2.5 cm
TerraStar-L	40 cm
RTK	1 cm + 1 ppm

Table 5: IMU performance.

Gyroscope		Accelerometer	
Technology	MEMS	Technology	MEMS
Dynamic range	400 °/s	Dynamic range	20 g
Bias instability	0.45 °/hr	Bias instability	0.075 mg
Angular random walk	0.06 °/√hr	Velocity random walk	0.06 m/s/√hr

3.2.3 Detection Model Development

A training dataset is created to test the detection framework using the setup explained in the Attack Detection Framework section. The driving route of the training dataset replicates an urban road structure, aiming to imitate real-world urban scenarios. The GNSS traces (latitude and longitude) are presented in **Figure 9**. Data from the IMU, including X, Y, Z acceleration, and X, Y, Z gyroscope data, along with speed data (north, east, and up velocity) and location data (latitude and longitude), are stored in a MongoDB database. The frequency of data is 1Hz. The Haversine formula is utilized to determine the distance traveled between two consecutive timestamps based on the data obtained from GNSS. (See **Equation 3**) (Robusto, 1957):

$$d = 2r\sin^{-1}\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\psi_2 - \psi_1}{2}\right)}\right) \quad (3)$$

where d represents the distance in meters between two points on the Earth's surface; r denotes the Earth's radius (6378 km); φ_1 and φ_2 represent the latitudes in radians; and ψ_1 and ψ_2 denote the longitudes in radians of two consecutive time stamps.

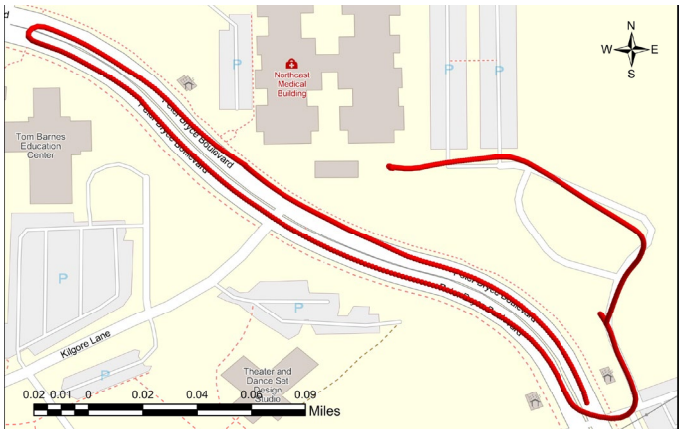


Figure 9: GNSS traces from the training dataset.

The prediction of the distance traveled by an AV between two consecutive timestamps is achieved using an LSTM architecture with 50 neurons (Khan *et al.*, 2020). The training and validation datasets include X, Y, Z acceleration, X, Y, Z gyroscope, and east, north, and up speed data. The output of the prediction model is the location shift between the current timestamp and the immediate future timestamp. The data generation frequency is 1 Hz, resulting in a time difference of 1 second between two consecutive timestamps. For training purposes, the continuous driving data from the training dataset (See **Figure 9**) is split into training with 7,193 observations and validation with 3,083 observations. Prior to feeding the sensor output into the LSTM training, the input features are normalized between 0 and 1. A Grid Search approach is employed to select the appropriate LSTM hyperparameters, such as the number of neurons, number of epochs, batch size, and learning rate due to the time series nature of the model (Khan *et al.*, 2020). The hyperparameter values and the optimizer's name of the LSTM model are listed in **Table 6**. After evaluating the LSTM-based prediction model, the Root Mean Square Error (RMSE) of the predicted location shift is measured to be 0.02 m, with the Maximum Absolute Error (MAE) being 0.06 m.

Table 6: LSTM model hyperparameters and optimizer.

Hyperparameters and Optimizer	Value
Number of neurons	50
Number of epochs	50
Batch size	12
Learning rate	0.01
Optimizer	Adam

Figure 10 displays the MAE loss profile, or learning curve, for the trained LSTM model. The y-axis represents the MAE loss values for both the training and validation datasets, while the x-axis indicates the

number of epochs. The learning curve illustrates that initially, the loss is comparatively high. However, the training loss steadily decreases and eventually stabilizes, indicating that the LSTM model is not under-fitted. Additionally, as both the training and validation losses reach a stable state quickly, it confirms that the LSTM model is not overfitted. Furthermore, the training and validation losses are consistently low, indicating the prediction model's effectiveness. The initial peak in both the training and validation losses indicates that the model was not well-generalized at that stage. However, as the number of epochs increases, the model becomes more stable and better generalized. It is worth noting that the training and testing data used in the analysis represent real-world driving data on urban routes. This ensures the neural network model demonstrates generalized behavior within an urban network context.

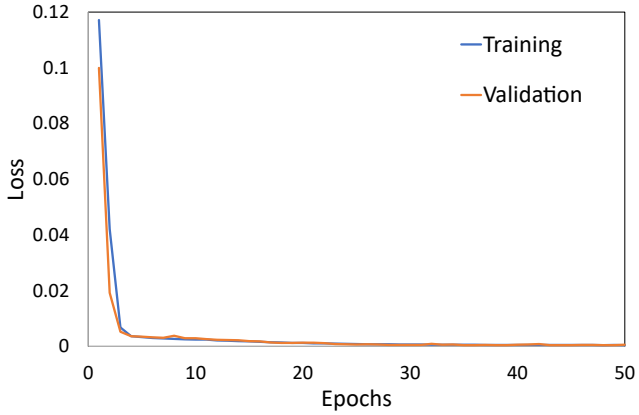


Figure 10: Comparison of Mean Absolute Error (loss) profiles with the optimal parameter set.

The RF statistical classifier is employed to categorize vehicle maneuvering into three classes: right turn, left turn, and no turn, based on gyroscope output. RF is preferred over KNN-DTW for real-time turn detection due to its superior efficiency with large datasets and faster prediction capabilities, essential for real-time processing. Furthermore, RF's robustness to noise and ability to handle complex, nonlinear relationships without extensive preprocessing made it a more practical choice for the unpredictable nature of GNSS data. The RF algorithm constructs classification trees using the dataset and aggregates predictions from all the individual trees to make the final classification decision. X, Y, and Z gyroscope data are used as the input features. The training dataset (10,026 observations) is labeled as a right turn, left turn, and no turn and used as a target label. The training dataset is not balanced in terms of left and right turn data. To solve this problem, Synthetic Minority Oversampling Technique (SMOTE) (Ramanishka *et al.*, 2018) is used to resample the dataset. The SMOTE sampling strategy is set to "auto" so that the minority class is oversampled to achieve an equal number of samples as the majority class. Then, the input feature data are normalized between 0 and 1 and fed to the RF model. Cross-validation is performed on the dataset by splitting it into five folds or subsets. The performance of the trained RF model is presented in **Table 4. 4** in terms of precision, recall, accuracy, and F1 score. Precision is the measure of how accurately a specific turn, or no turn, is detected out of all observations. The precision of the RF model varies from 93% to 96%. Recall refers to the percentage of the observations where the classification is correct. The recall varies from 89% to 97%. The classification accuracy is 94%. The F1 score reflects the balance between precision and recall. The F1 score ranges from 0.92 to 0.95, which proves that the precision and recall are well-balanced.

Table 7: Random Forest model validation result.

Turn type	precision	recall	accuracy	f1-score
Right	0.93	0.95	0.94	0.94
Left	0.93	0.97	0.94	0.95
No turn	0.96	0.89	0.94	0.92

To detect turns using GNSS data, three consecutive latitude and longitude data to determine the turn angle and based on the turn angle value the turns are classified. The vehicle state is determined based on the SPAN data, which consists of north, east, and up velocity components. The RMSE of the SPAN velocity is measured to be 0.03 m/s. If any of the three velocity components (north, east, or up) exceed the threshold of 0.03 m/s, the vehicle state is classified as "moving." Otherwise, if all velocity components are below or equal to 0.03 m/s, the vehicle state is classified as "standstill."

All the pre-trained models operate in real-time on the onboard computing unit during a trip. The detection algorithms run in parallel. Data from GNSS, IMU, and speedometer are stored in the MongoDB database. Each model utilizes the required real-time data for spoofing detection. For instance, at timestamp t, the GNSS and IMU data are immediately fed to the distance traveled determination model to calculate the actual distance traveled between timestamp t and t-1. Simultaneously, the distance traveled prediction model, which is a pre-trained LSTM, predicts the distance traveled between timestamp t and t-1. The results from both models are compared to verify if the GNSS signal is being spoofed. In parallel, the turning maneuver detection model leverages the gyroscope data at timestamp t to classify the maneuver type. This information is cross-referenced with the turn detection model using data from timestamps t-2, t-1, and t to detect any signs of a spoofing attack in real-time. Furthermore, the vehicle's motion state is determined using the SPAN data at timestamp t. This information is then compared with the GNSS speed output to detect any potential attack in real-time.

3.2.4 Attack Scenarios

Figure 11 presents two types of GNSS spoofing scenarios (i.e., turn-by-turn and wrong turn) generated in this validation study. **Figure 12** presents example spoofing attack scenarios. The spoofing detection model introduced in this chapter is at the navigation solution level and solely dependent on the latitude and longitude data from the GNSS, and the detection model does not require considering GNSS signal parameters. Therefore, by substituting the legitimate latitude and longitude data in the database where the GNSS location data is stored in real-time with spoofed latitude and longitude values (See **)** to mimic a spoofing attack and the detection framework's performance against such attacks can be effectively tested. For this purpose, a spoofed database, which contains the latitude and longitude of spoofed routes for spoofing scenarios, is created. The spoofed route is selected based on the assumption that an attacker possesses information about the probable route of the victim AV before initiating the attack. This knowledge allows the attacker to gradually manipulate the perceived location of the AV, which is challenging to detect. To create and obtain latitude and longitude data along the route, ArcGIS's Network Analyst tool is utilized. To replicate the movement of an actual vehicle, the update rate of the spoofed route's location is determined based on the average speed observed on that specific roadway. For the testing routes, it is assumed that the average vehicle speed is 25 mph, equivalent to 11.18 m/s. Considering the AV's GNSS data output frequency to be 1 Hz, latitude and longitude points along the spoofed route are generated approximately 11 meters apart from each other. In order to generate a spoofing attack, the latitude and longitude data columns in the database are substituted with the spoofed route data from the spoofed database. Consequently, the AV will perceive the spoofed location as authentic. However, it is important to note that during the attack, all other sensor data remains uncompromised. This means that all other data columns are continuously updated in real-time with legitimate data, maintaining their integrity throughout the spoofing attack.

A total of three spoofed routes are generated for each spoofing attack scenario. A route is created for the turn-by-turn attack data wherein the AV's location is shifted from its current position. The shift is directed towards the adjacent parallel road. As a result, the shift is intentionally kept relatively small. To create wrong turn attack, six intersections are chosen, three wrong right turns, and three wrong left turns routes are created. Whenever the AV is near one of the chosen intersection, spoofed route is injected to the

database to execute the attack. In every instance of spoofing attacks, the database is compromised via an SQL injection technique, which permits the attacker to manipulate the database by adding, deleting, or altering its contents (D'Antonio *et al.*, 2011; Almutairy *et al.*, 2023).

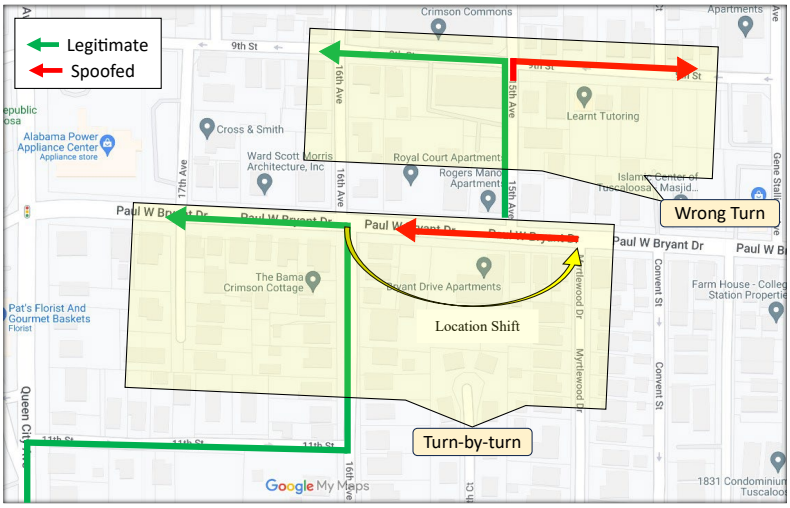


Figure 11: Spoofing attack types.

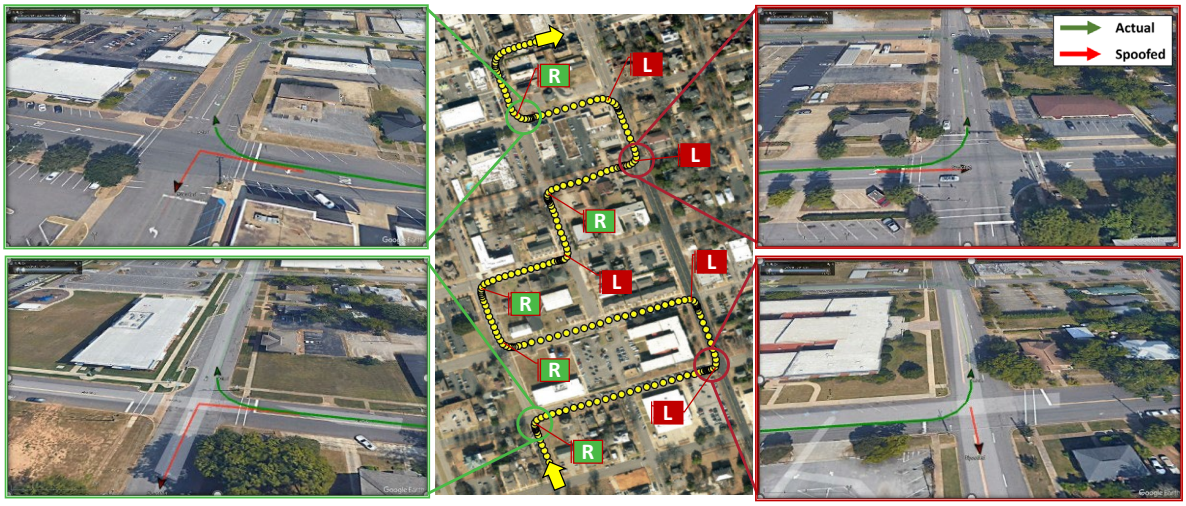


Figure 12: Sample attack scenarios.

3.3 GIS and Landmark-Based Navigation System for Autonomous Vehicles

3.3.1 Navigation Framework

This sub-section introduces a comprehensive navigation framework (See **Figure 13**) that combines Geographic Information System (GIS) and landmark-based techniques to provide dependable navigation for AVs in GNSS-challenged urban landscapes. It focuses on harmonizing the vehicle's integral systems to achieve navigation accuracy. The framework employs a dual-strategy approach: first, it utilizes a fusion of in-vehicle sensor data—including GNSS, accelerometer, gyroscope, speedometer, and camera—to maintain precise navigation of the AV, particularly when GNSS signals are unreliable. This fusion does not rely on external signals. The two-fold strategy involves (i) determining the AV's position along the planned route by estimating the distance traveled at each timestamp, and (ii) refining the AV's location by recognizing and utilizing landmarks along the route for positional correction.

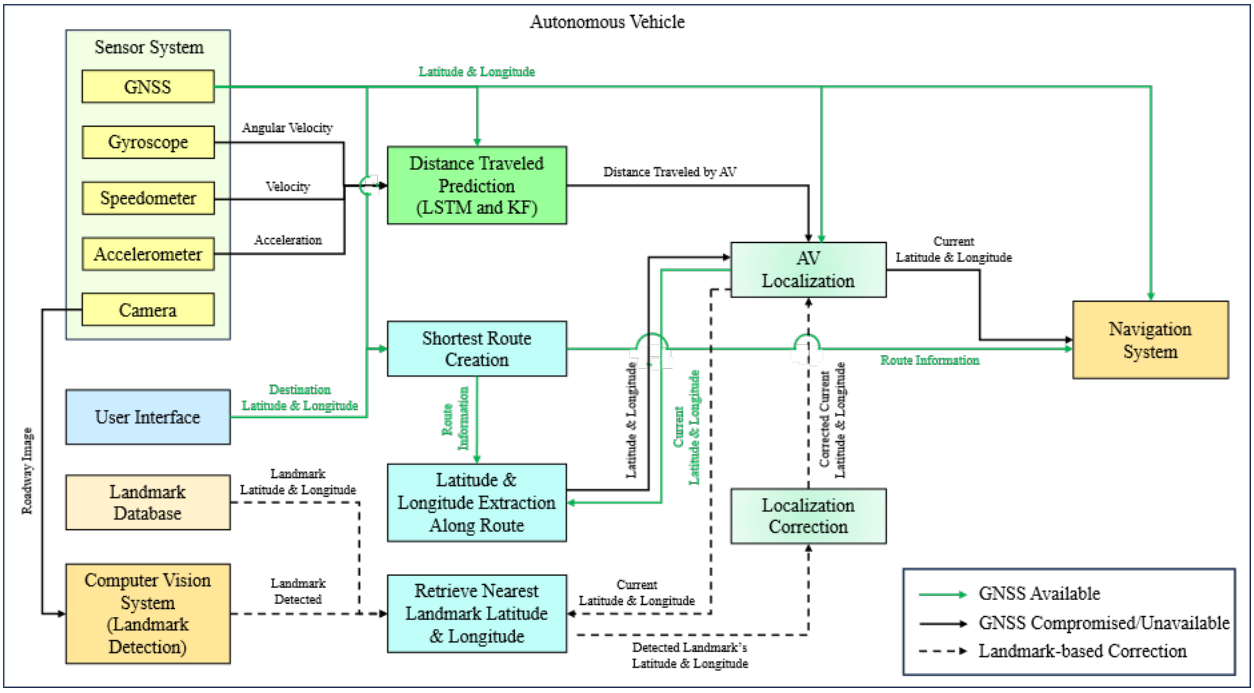


Figure 13: GIS and landmark aided GNSS-less navigation framework.

The goal of the first strategy is to first find the shortest route between the AV's current location and the destination and then predict the distance travelled by the AV in each timestamp using both machine learning (ML) model and KF. Long Short-Term Memory (LSTM) network, a specialized form of a recurrent neural network (RNN) known for its effectiveness in time-series prediction tasks is used for the distance traveled prediction. The LSTM network is particularly suitable for this scenario due to its ability to remember and process long-term dependencies in time-series data. To train the model, before equipping the AV with the presented framework, data from uncompromised GNSS, accelerometer, gyroscope and speedometer are used. Further a KF is also used to estimate the distance traveled by the AV using speedometer and accelerometer data. Based on the predicted distance value the location of the AV is updated along the route.

The drawback of the first strategy is, it accumulates error over time, necessitating correction for reliable AV navigation. Conversely, the second strategy mitigates this issue by leveraging readily identifiable landmarks for corrections. In this chapter traffic intersections and big parking lot entry and exits are used as landmarks. The accurate location of the landmarks is stored in a landmark database. Given the AV's reliance on a detailed and precise map for navigation, it is reasonable to assume that this map accurately represents these landmarks. The computer vision system of AV should be able to detect the landmarks and whenever a landmark will be detected the navigation system will correct the AV location using the accurate location of the landmark. By using these two strategies the AV can not only locate itself whenever it detects a landmark but also in between two landmarks ensuring consistent and reliable navigation.

At the beginning of a trip, it is assumed that GNSS is not compromised, and the AV Shortest Route Creation module receives the current location of the AV from the GNSS and the destination as a user input using the User Interface module. During the trip, when GNSS is not compromised, the navigation AV's navigation system receives the current location (latitude and longitude) directly from GNSS module and the route information and turn by turn navigation instructions from the Shortest Route Creation Module. Latitude & Longitude Extraction Along Route module extract latitude and longitude along the route and AV's location is updated along these extracted points by AV Localization module using current location data from GNSS. All the communications during the GNSS uncompromised situation are presented in green line in **Figure**

13. After an event of GNSS outage or attack is detected, the AV Localization module starts feeding the current latitude and longitude to the Navigation System instead of GNSS using the strategy 1 and strategy 2. Strategy 1 & 2 are described in detail in the following subsections.

3.3.1.1 Development of Strategy 1 : Localization

The first strategy is presented in detail in **Figure 14**. The first strategy includes the Shortest Route Creation, Latitude & Longitude Extraction Along Route, Distance Traveled Prediction and AV Localization modules. The output of this strategy is the current location in terms of latitude and longitude. Each module is explained in detail in the following.

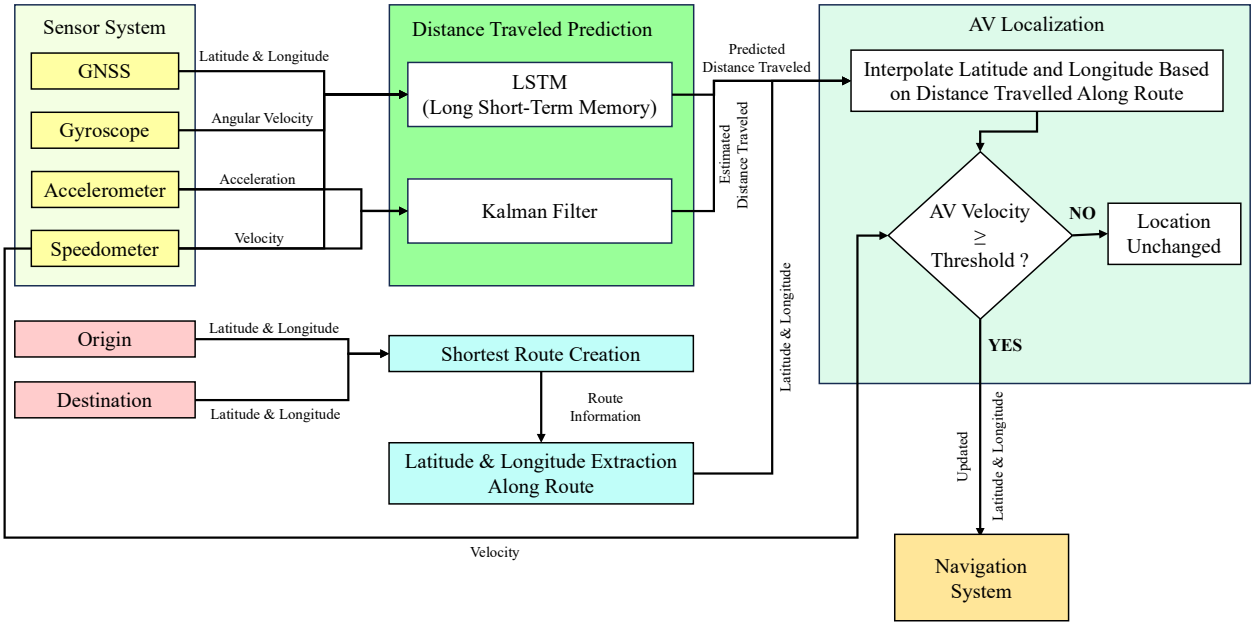


Figure 14: Strategy 1 for localizing AV without GNSS.

3.3.1.1.1 Shortest Route Creation and Latitude & Longitude Extraction Along Route

The AV is equipped with a detailed and accurate map and the road network is presented as graph of roads which is a subset of drivable roads. Roads are represented as links and traffic intersections are presented as nodes. At the beginning of a trip, current (origin) location (latitude and longitude) fed to this module from the GNSS, and the destination is fed as an address from the user interface. From the address the destination latitude and longitude are extracted using ArcGIS "geocode" function that convert address into geographic coordinates. Specific address is used as otherwise the "geocode" function would return multiple results. Form the current and destination latitude and longitude the route is created using Open-Source Routing Machine (OSRM), a sophisticated tool designed for calculating optimal driving routes using road network data. From the provided coordinates, OSRM calculates the most efficient route for driving, considering various factors like road types, turn restrictions, and traffic conditions (if data is available). It uses advanced algorithms like Contraction Hierarchies for fast and efficient route computation. Detailed representation of the route is used ensuring that the entire route is comprehensively covered setting parameter "overview=full." In the latitude and longitude extraction along route module, an interpolation method is employed and 'geopy' python library is used. This method calculates additional points between the given route coordinates at specified intervals. In this case, the interval is set to 1cm. The interpolation is done by

linearly distributing points between each pair of successive coordinates along the route, thereby generating a dense and continuous sequence of latitude and longitude points that accurately trace the path.

3.3.1.1.2 Distance Traveled Prediction

This section describes how the module predicts the distance traveled by the AV at every timestamp, utilizing onboard sensors such as the Gyroscope, Accelerometer, and Speedometer. This chapter employs two techniques for predicting the distance traveled: (i) LSTM and (ii) KF. The LSTM model is trained with data from uncompromised GNSS. Detailed explanations of both models are provided below.

3.3.1.1.2.1 LSTM-based Prediction

A two-stacked LSTM network is used for predicting the distance traveled by AV at each timestamp. The LSTM model is trained with speedometer, gyroscope (Gyro) and accelerometer (Accl). The velocity from the speedometer is 1D horizontal velocity. Gyroscope and accelerometer provide angular velocity and acceleration respectively in three dimensions (x, y, z). Hence the input layer has seven inputs. The hyperparameter values and optimizer are presented in **Table 8**. The LSTM network's architecture (See **Figure 15**) begins with a foundational layer consisting of 64 units. These units, known as memory cells, are adept at capturing and learning long-term dependencies within the dataset. This ability is crucial for understanding the temporal dynamics inherent in time series data. To mitigate the risk of overfitting, a Dropout layer with a rate of 0.2 is integrated immediately following the initial LSTM layer. The purpose of this layer is to randomly deactivate a fraction of the input units during the training phase, enhancing the model's capacity to generalize beyond the training data. This is a critical step in ensuring that the model performs well on new, unseen data. The network then incorporates a second LSTM layer, also with 64 units. This layer builds upon the foundational layer, allowing the model to further refine its understanding of the data's underlying patterns. Following this, another Dropout layer with the same rate of 0.2 is added. The repeated application of Dropout layers throughout the network serves as a robust defense mechanism against overfitting, ensuring the model's predictive reliability.

The architecture culminates in a Dense layer, which contains a single neuron. This layer is responsible for producing the final output of the network, effectively translating the learned features and patterns into a tangible prediction. The output layer consists of single cell which is the predicted distance traveled. For the model's compilation, the Adam optimizer is employed due to its effectiveness in handling sparse gradients and adaptive learning rates. The optimizer's learning rate is set to 0.001, a value carefully chosen to balance the speed of learning with the stability of the algorithm. The Mean Squared Error (MSE) is selected as the loss function, a standard choice for regression tasks. MSE is particularly effective in quantifying the accuracy of the model when predicting continuous variables. In terms of training data and preprocessing, the model utilizes speed, acceleration, gyroscope data and distance travelled data that is calculation based on the GNSS output. The data is synchronized to 10Hz to ensure consistency and uniformity in the input data fed into the model.

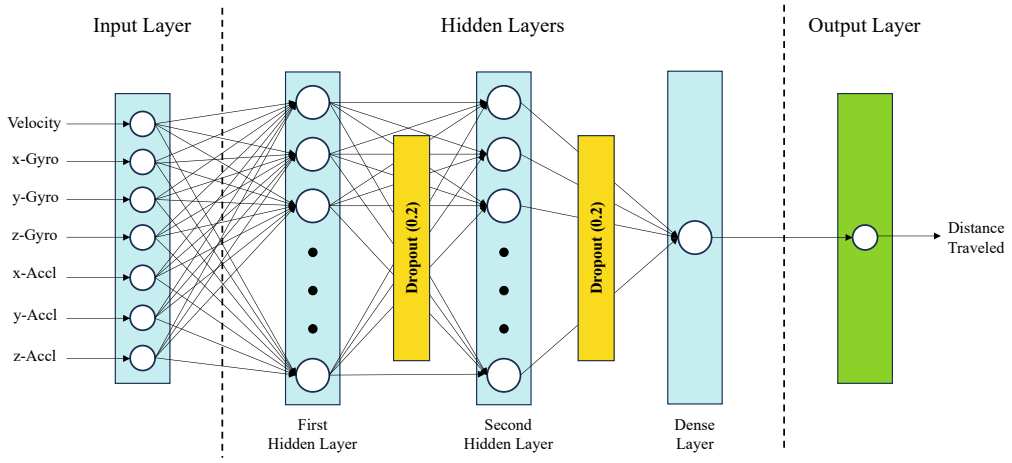


Figure 15: LSTM network architecture.

Table 8: Prediction model (LSTM) hyperparameters and optimizer.

Hyperparameters and Optimizer	Value
1st LSTM Layer Units	64
1st LSTM Layer Dropout	0.2
2nd LSTM Layer Units	64
2nd LSTM Layer Dropout	0.2
Dense Layer Units	1
Number of Epochs	50
Batch Size	32
Optimizer	Adam
Learning Rate	0.001

3.3.1.1.2.2 Kalman Filter-based Estimation

In order to predict the distance between each timestamp, a dynamical model based on the velocity and acceleration data provides a baseline approach. Since IMU data is prone to sensor errors, bias, and drift it is necessary to filter out the effects of these in order to identify the optimal distance moved. In this regard, KF based implementations are a popular choice. Hence, in our approach a KF based distance estimation is performed to obtain an optimal estimate of the distance.

3.3.1.1.2.2.1 Dynamics Model

The dynamics model is used to predict the state in the next time step for the AV. In this application, the state comprises of the vehicle's distance moved, horizontal velocity and the acceleration component along the forward direction of the vehicle's body. Using the equations of motion, the position and velocity for the next timestep were predicted using the data. Acceleration was assumed to be constant between each time step update. Therefore, the state vector for the system may be described as:

$$\mathbf{X_k} = [x_k, v_k, a_K]^T \tag{4}$$

Where x_k is the distance moved in the previous timestep, v_k is the velocity of the vehicle and a_k is the acceleration of the vehicle.

Based on these state variables, the following equations are used to describe the system dynamics and predict the next state:

$$x_{k+1} = v_k \Delta t + \frac{1}{2} a_k \Delta t^2 \quad (5)$$

$$v_{k+1} = v_k + a\Delta t \quad (6)$$

$$a_{k+1} = a_k \quad (7)$$

This is stated in vector form as:

$$\mathbf{X_{k+1}} = \mathbf{F X_k} + \mathbf{u} \quad (8)$$

Where $\mathbf{u}$ is the process noise and $\mathbf{F}$ is the state transition matrix given by:

$$\mathbf{F} = \begin{pmatrix} 0 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{pmatrix} \quad (9)$$

where Δt is the time interval between updates and is depended on the rate of the IMU data.

3.3.1.1.2.2.2 Observation Model

The observation model maps the state of the vehicle to the sensor measurements. In this application, the measurements are the horizontal velocity and the acceleration component along the forward direction of the vehicle's body. The horizontal velocity of the vehicle is obtained from the doppler shift velocity of the Novatel GNSS receiver, which measures the velocity of the receiver antenna with respect to the ground. Although this horizontal velocity may not be aligned with the forward direction of the vehicle's body, the discrepancy would be negligible within the small sampling time of the data. Hence, it is assumed that the horizontal velocity can be used in place of the forward velocity in the vehicle's body frame. The velocity derived from wheel speed sensors or odometer readings would be more appropriately aligned with the body frame. In practice however, discrepancies within the wheel radius due to wear and tear and other factors such as wheel slip would induce errors within the wheel speed sensors or odometer readings, so our assumption would not adversely affect performance as compared to actual body frame aligned velocity. Data for the acceleration is obtained from the IMU sensor. The IMU is able to measure three axis acceleration along its own frame. During the data collection, the IMU unit was placed inside the vehicle in a fixed position, with its x direction aligned along the vehicle's front. Thus, the x component of the acceleration was used as the measure of the vehicle's acceleration. The measurement vector is represented as:

$$\mathbf{z_k} = [v_k, a_k]^T \quad (10)$$

The observation model is used to map the state to the measurement as follows:

$$\mathbf{Y_k} = \mathbf{H X_k} + \mathbf{w_k} \quad (11)$$

Where $\mathbf{w}$ is the measurement noise and $\mathbf{H}$ is the observation matrix given as:

$$\mathbf{H} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (12)$$

3.3.1.1.2.2.3 Kalman Filter

The Kalman Filter consists of two stages which are the prediction and correction stages.

Prediction stage: In the prediction stage, **Equation 13** predicts the next state of the system ($\hat{\boldsymbol{x}}_{\boldsymbol{k}}^{-}$) based on the previous estimated state ($\hat{\boldsymbol{x}}_{\boldsymbol{k-1}}^{-}$) and control inputs ($\boldsymbol{u_k}$), incorporating the effects of process noise ($\boldsymbol{Q}$). It accounts for the natural progression of the system over time. This equation updates the state covariance ($\boldsymbol{P}_{\boldsymbol{k}}^{-}$), which quantifies the estimated accuracy of the state predictions. It scales the uncertainty of the previous state estimate with the process noise to predict the uncertainty of the current state estimate.

$$\hat{\boldsymbol{x}}_{\boldsymbol{k}}^{-} = \boldsymbol{F}\hat{\boldsymbol{x}}_{\boldsymbol{k-1}}^{-} + \boldsymbol{u_k} \quad (13)$$

$$\boldsymbol{P}_{\boldsymbol{k}}^{-} = \boldsymbol{F}\boldsymbol{P}_{\boldsymbol{k-1}}\boldsymbol{F}^{\boldsymbol{T}} + \boldsymbol{Q} \quad (14)$$

Where **Q** is the process covariance matrix.

Correction stage: The correction stage incorporates the new measurement data to refine the system's state estimate. This stage adjusts the predicted state based on the difference between the actual measurements and the predicted measurements. The Kalman Gain ($\boldsymbol{K_k}$) is computed using **Equation 15**, balancing the estimated state uncertainty and measurement uncertainty. It determines how much the predictions should be corrected based on the new measurements. **Equation 16** equation updates the predicted state ($\hat{\mathbf{x}}_{\mathbf{k}}^{-}$) with the discrepancy between the actual measurement ($\mathbf{z_k}$) and the predicted measurement ($\boldsymbol{Y}_{\boldsymbol{k}}^{-}$), weighted by the Kalman Gain. This produces a more accurate estimate of the state ($\hat{\mathbf{x}}_{\mathbf{k}}$). Finally, the state covariance ($\mathbf{P_k}$) is updated using **Equation 17** to reflect the reduced uncertainty after incorporating the measurement. This step adjusts the estimate of the system's uncertainty, typically lowering it in light of the new information.

$$\boldsymbol{K_k} = \boldsymbol{P}_{\boldsymbol{k}}^{-}\boldsymbol{H}^{\boldsymbol{T}}\left(\boldsymbol{H}\boldsymbol{P}_{\boldsymbol{k}}^{-}\boldsymbol{H}^{\boldsymbol{T}} + \boldsymbol{R}\right)^{-1} \quad (15)$$

$$\hat{\mathbf{x}}_{\mathbf{k}} = \hat{\mathbf{x}}_{\mathbf{k}}^{-} + \mathbf{K_k}(\mathbf{z_k} - \boldsymbol{Y}_{\boldsymbol{k}}^{-}) \quad (16)$$

$$\mathbf{P_k} = (\mathbf{I} - \mathbf{K_k}\mathbf{H})\mathbf{P}_{\mathbf{k}}^{-} \quad (17)$$

By repeatedly cycling through the prediction and correction stages, the KF efficiently merges predictions based on the system's dynamics with incoming measurements, minimizing the error in the state estimates.

3.3.1.1.3 AV Localization

The AV Localization module receives input in the form of the estimated distance traveled and a sequence of latitude and longitude points from the Route-based Latitude & Longitude Extraction process. Utilizing the most recent position and the computed distance traversed, it calculates an interpolated new location, ensuring that the AV remains on the pre-determined shortest path initiated at the start of the journey. However, it's crucial to halt location updates when the AV comes to a stop. Despite being stationary, calibration discrepancies in the speedometer or disturbances from electrical noise might lead to inaccurate distance readings greater than zero. To mitigate such discrepancies, the system is configured to update the vehicle's location only when the speedometer's velocity reading is at or above a certain threshold. This threshold is established through experimental trials with the vehicle to ascertain and accommodate the actual measurement error.

3.3.1.2 Development of Strategy 2 : Correction

The correction system mimics a landmark-based navigation system (See **Figure 16**). The landmarks used in this framework are traffic intersections and large parking lot entry and exists. The motivation behind

selecting the intersections as the landmarks is intersections are easy to detect and classify. A traffic intersection can be signalized, unsignalized and uncontrolled. Signalized intersections are equipped with traffic lights, and they are easiest to detect by the computer vision system of the AV. Roundabouts, stop sign-controlled, yield sign-controlled and uncontrolled. The landmark database in the framework consists of the locations (latitude and longitude) of the intersections and large entry/exit along with the type of the intersection. That is at the beginning of the trip the AV will have a shortest route with latitude longitude information along the route along with the location and types of intersections along the route. As the AV detects intersections and parking entry/exits by default which is one of its core capabilities, it is assumed that using the camera and computer vision system AV will be able to detect and classify the landmarks. After detecting a landmark, the retrieve nearest landmark latitude & longitude module finds the nearest landmark from the current location that comes from the AV localization module. The landmark location is then fed to the localization correction module, and the current location is replaced with the landmark location. It is assumed that for AV navigation purpose, the database contains accurate location and classification data of the landmarks. Hence the presented framework can guide the AV to its destination during GNSS outage or compromised situation making the AV travels safe and secure.

It is assumed that the core AV systems include propulsion system, sensor system (comprising GNSS, IMU, cameras, RADAR, and LIDAR), control system, computing system, power supply, user interface and additional peripherals vital for navigation tasks. Sensor fusion algorithm, computer vision system and navigation system and obstacle avoidance system are part of the AV control system. The sensor system serves as the backbone of the AV's operational ecosystem, with each component playing a vital role. The GNSS provides global positioning data, while the IMU offers crucial data on vehicle motion, including acceleration and rotation. Cameras, RADAR, and LIDAR collectively contribute to a 360-degree perception of the vehicle's surroundings, essential for detecting obstacles, road signs, and other critical environmental features. Sensor fusion algorithms integrate data from various sensors to create a coherent understanding of the vehicle's environment. Computer vision systems process visual data for object recognition, while sophisticated navigation algorithms facilitate efficient pathfinding and incorporate obstacle avoidance strategies.

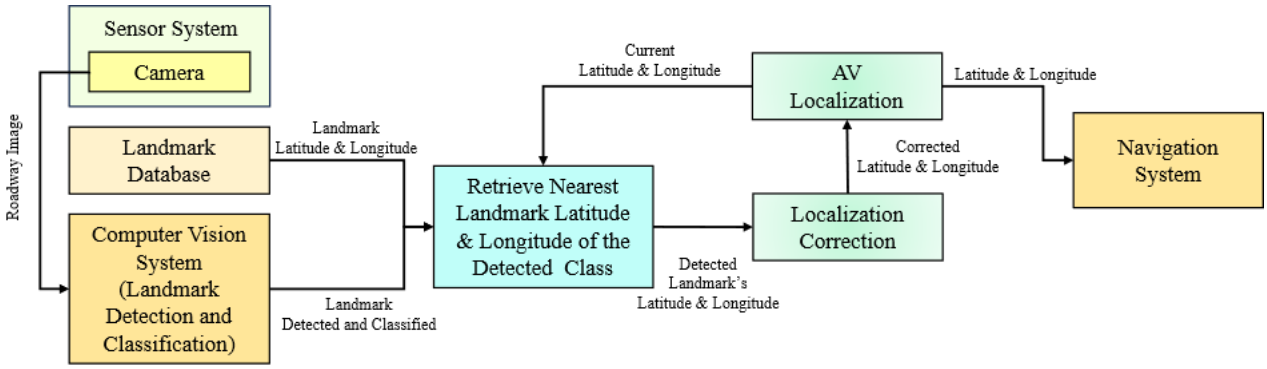


Figure 16: Landmark based correction.

3.3.2 Experiments

The effectiveness of the proposed framework is evaluated through experiments. The following section provides a comprehensive overview of the experimental architecture, the setup of the experiments, and the meticulous preparation of the data involved.

3.3.2.1 Experimental Framework

The experimental setup is detailed in **Figure 17** which is adopted from the framework presented in the Navigation Framework section, where a predefined route is navigated by a vehicle equipped with sensors commonly found in AVs. A conventional car, driven by a human, is augmented with these sensors to conduct the experiments. The vehicle's exact position at any moment is deduced from the distance it covers, with predictions made at every timestamp. These predictions are facilitated by a dual-layer LSTM network, outlined in the LSTM-based Prediction subsection, and a Kalman Filter, described in the Kalman Filter-based Estimation section. Training for the distance prediction module incorporates data from GNSS, gyroscope, accelerometer, and speed, the latter calculated from GNSS data over time to produce an average speed rather than instantaneous velocity.

The process begins by inputting the start point's coordinates and the destination address into the shortest route creation function, with the destination's coordinates obtained via the ArcGIS "geocode" function, which translates addresses into geographic coordinates. The route is then generated using the OSRM, based on these coordinates. To ensure the route's precision, the latitude and longitude extraction module uses interpolation, facilitated by the 'geopy' Python library, to calculate additional points at roughly 1cm intervals along the route. This creates a detailed and continuous sequence of coordinates that closely follows the planned path.

Location accuracy is further enhanced using a landmark database and landmark detection timestamps. This database holds precise locations of landmarks, and the timestamps when the vehicle detects these landmarks are recorded. Assuming the AV's computer vision system can identify and classify these landmarks accurately, the moment a landmark is no longer visible marks the timestamp when the vehicle passes it. During landmark detection, the nearest landmark is identified, and the vehicle's location is corrected to the nearest point on the route to the detected landmark. This correction ensures that the vehicle's location is continuously updated and aligned with the actual route at every landmark, feeding into the predicted location module for real-time adjustments.

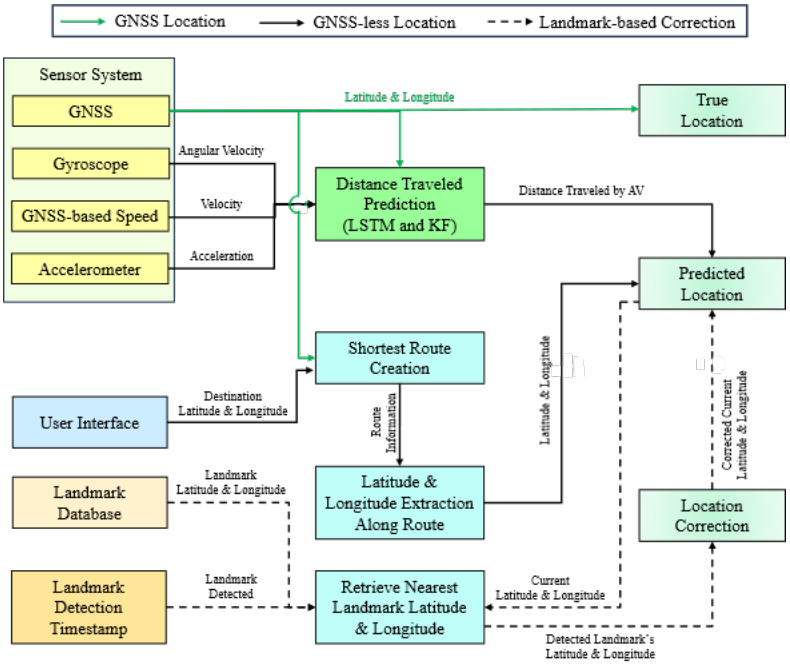


Figure 17: Experimental framework.

3.3.2.2 Experimental Setup

The experimental setup (See **Figure 18**) is designed to evaluate the integrated performance of the landmark and GIS-aid navigation system framework presented in this chapter. Central to the configuration is the Single-antenna CPT7700 system from NovAtel, a compact solution that combines a high-precision GNSS receiver with an inertial measurement unit (IMU). CPT7700 contains an OEM7700 multi-frequency GNSS receiver, which can track current and upcoming GNSS constellations, including GPS, GLONASS, Galileo, BeiDou, QZSS, and IRNSS. The CPT7700 is versatile, offering SPAN GNSS+INS capability. NovAtel's SPAN technology combines precision GNSS positioning with inertial navigation to provide a robust, continuously available, position, velocity, and attitude solution that can function in GNSS-denied environments. The performance of the GNSS receiver is being presented in **Table 9**. CPT7700 is also equipped with the high-performing Honeywell HG4930 Micro Electromechanical System (MEMS) IMU containing gyroscope and accelerometer. The performance of the IMU is being presented in **Table 10**. A CPT7 I/O2 cable is being used to connect CPT7700 with a Windows-based onboard computing unit with the USB port, which supports a hi-speed (480Mb/s) data rate. The Hexagon VEXXIS® GNSS-850 tough high-precision antenna with superior tracking performance is being used, featuring multi-point feeding network and radiation pattern optimization technology. This antenna is compatible with a wide range of satellite systems, including GPS, GLONASS, BeiDou, and Galileo. It has the ability to track low-elevation satellites while maintaining a high gain for higher-elevation satellites, making it suitable for applications where the sky is partially visible, such as operating close to tree lines, under foliage, or in urban canyons. CPT7700 is capable of internal data logging up to 16 GB. The system's IMU data rate can be upgraded to 400 Hz, ensuring high-frequency motion detection. For the experiments, TerraStar-L correction services are utilized and IMU data are logged at 100Hz frequency.

Table 9: GNSS receiver performance.

Positioning	Accuracy (RMS)
Single Point L1	1.5 m
Single Point L1/L2	1.2 m
SBAS	60 cm
DGPS (code)	40 cm
TerraStar-C PRO	2.5 cm
TerraStar-L	40 cm
RTK	1 cm + 1 ppm

Table 10: IMU performance.

Gyroscope		Accelerometer	
Technology	MEMS	Technology	MEMS
Dynamic range	400 °/s	Dynamic range	20 g
Bias instability	0.45 °/hr	Bias instability	0.075 mg
Angular random walk	0.06 °/√hr	Velocity random walk	0.06 m/s/√hr

To record the video along the route Stereolabs ZED 2 camera is employed, a cutting-edge stereo vision system. It captures high-definition 3D video and offers video output resolutions up to 2K at 15 FPS, with lower resolutions available at higher frame rates. Its motion sensors operate at a data rate of 400Hz, and its object detection capabilities allow for tracking of persons and vehicles with a range of up to 20 meters in 3D and up to 40 meters in 2D. The instruments are integrated into a sedan, which serves as the mobile platform for our experiments. This vehicle is outfitted with the necessary hardware interfaces to support data acquisition and processing. An onboard laptop with a wireless internet connection runs Python-based applications and interfaces with ArcGIS Online and OpenStreetMap via an API key. Video data is stored on the laptop's internal memory. A Jackery Portable Power Station Explorer 240, 240Wh Backup Lithium Battery, 110V/200W Pure Sine Wave AC Outlet is used as power supply.

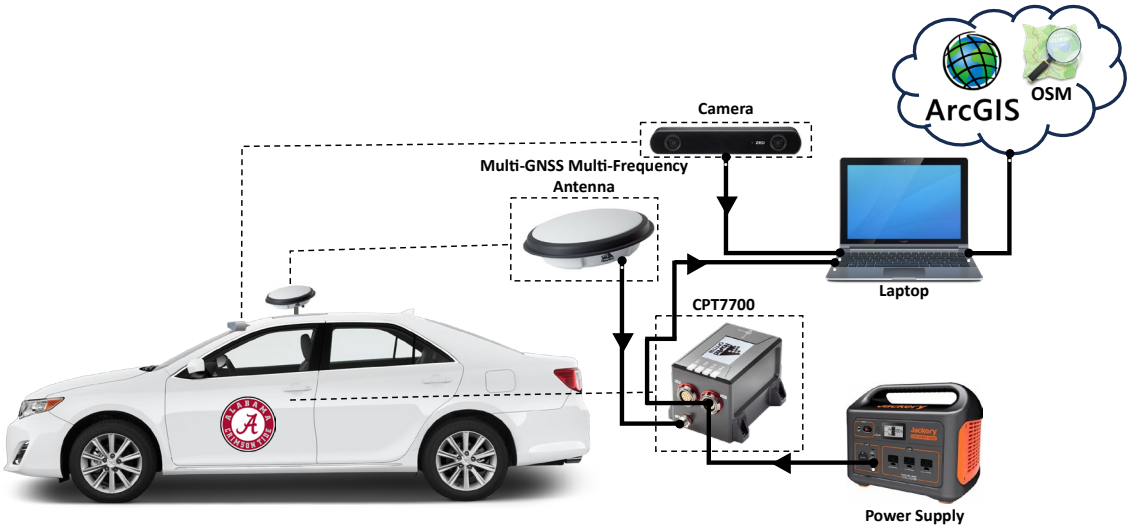


Figure 18: Experimental setup.

3.3.2.3 Data Preparation

In the conducted study, a specific roadway corridor shown in **Figure 19** is delineated for analysis. This corridor, extending over a distance of approximately 1.7 miles, is strategically selected due to its encapsulation of various urban traffic elements and conditions. The designated route is characterized by a multitude of signalized intersections, encompassing both vehicular and pedestrian traffic controls. It features an array of intersection typologies, including both signalized and unsignalized intersections, further contributing to its complexity. Along this route, there are numerous parking facility entry points, which introduce additional variables in traffic flow and control. Furthermore, the route is notable for containing several left-turn maneuvers. In total, this urban transportation corridor comprises seven signalized intersections, seven stop-sign controlled intersections, and nine distinct parking lot access points.

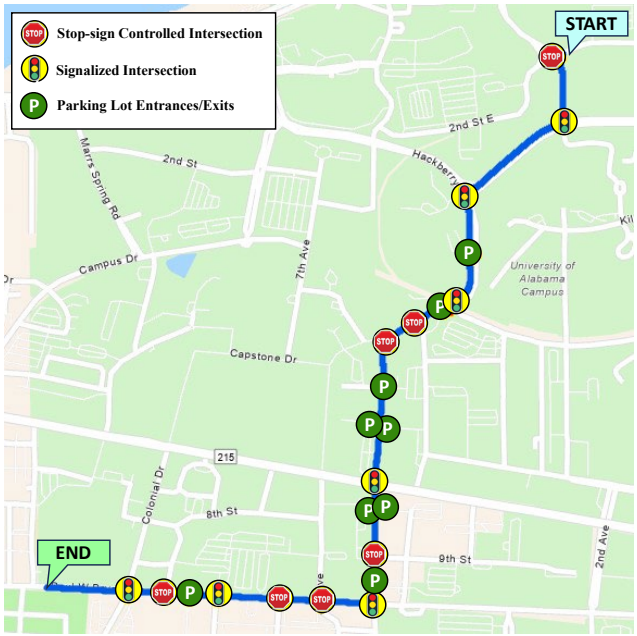


Figure 19: Route used during experiments.

Location coordinates are acquired at a frequency of 20Hz utilizing the "BESTPOS" log, which records the optimal position as determined by the receiver. This log captures both latitude and longitude in degrees, earmarking these coordinates for subsequent analytical procedures. In the absence of direct speedometer readings, the primary source of velocity data is derived from the "BESTGNSSVEL" log, which provides the best available GNSS velocity measurements. The update rate for this velocity data is set at 2Hz, resulting in a velocity latency of approximately 0.25 seconds. Typically, this velocity is calculated based on either the average change in pseudorange during the defined interval or through the application of the Real-Time Kinematic (RTK) Low Latency filter. Additionally, raw IMU data, encompassing X, Y, Z axes accelerations (indicative of velocity changes) and gyroscopic data (reflecting angular changes), is recorded at a high frequency of 100Hz via the "RAWIMU" log. To convert these raw IMU measurements into practical units, the acceleration and gyroscope data are scaled by factors of 1.862645149230957e-09 and 1.1641532182693481e-10 respectively, and then multiplied by the data recording frequency of 100Hz. This scaling process translates the measurements into acceleration units of m/s^2 and gyroscope readings in rad/s. In order to ensure temporal alignment of the data sets from the various sensors, a synchronization process is employed. This process uses the GNSS location timestamp as a reference point and aligns the nearest corresponding data points from the other sensors, namely the IMU and GNSS velocity measurements. Consequently, the synchronized dataset adheres to a uniform frequency of 20Hz, ensuring consistent temporal intervals across all data types. Sample data is presented in **Figure 20**.

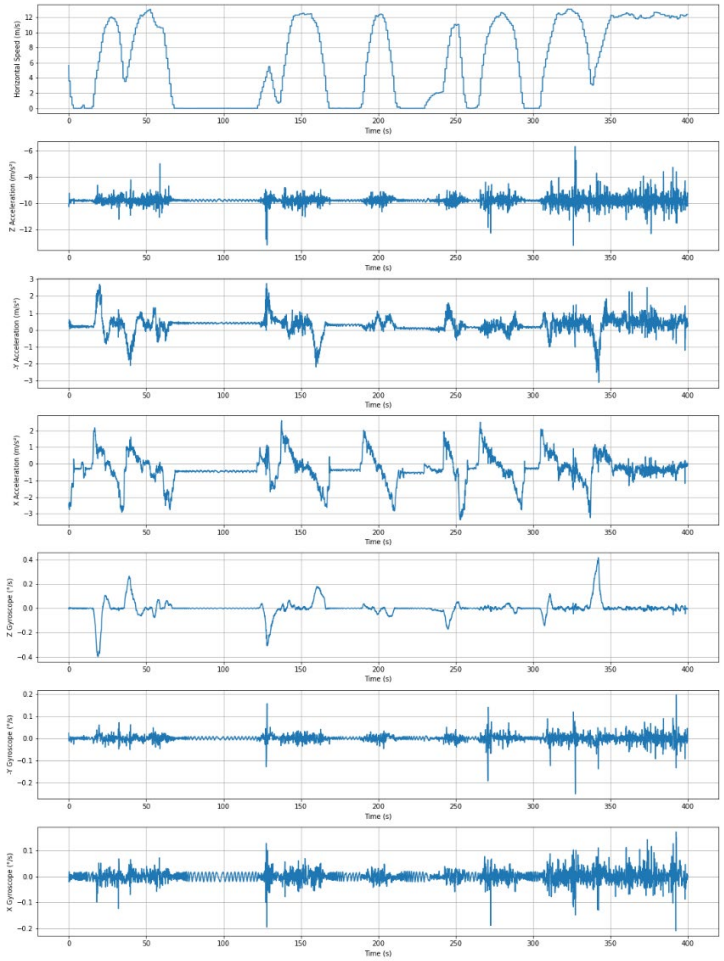


Figure 20: Sample IMU and speed data.

The Haversine formula is utilized to determine the distance traveled between two consecutive timestamps based on the data obtained from GNSS. (See **Equation 18**):

$$d = 2r \sin^{-1}\left(\sqrt{\sin^2\left(\frac{\varphi_2 - \varphi_1}{2}\right) + \cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\psi_2 - \psi_1}{2}\right)}\right) \quad (18)$$

where d represent the distance in meters between two points on the Earth's surface; r denotes the Earth's radius (6378 km); φ_1 and φ_2 represent the latitudes in radians; and ψ_1 and ψ_2 denote the longitudes in radians of two consecutive time stamps. This distance is further used for training the distance prediction model.

3.3.2.4 Route Creation and Feature Extraction

Python libraries such as Pandas (for data manipulation), GeoPandas (for handling geospatial data), OSMnx (for OpenStreetMap data manipulation), Folium (for map rendering), NetworkX (for graph-theoretic analysis), Matplotlib (for plotting), and NumPy (for numerical computations) are used for the route creation and feature extraction along the route. Additionally, mathematical functions were imported from the 'math' library to facilitate geospatial calculations. The python script created for performing the application takes the origin and destination address from the user as input. Then using our "get_lat_lon" function the latitude and longitude of the addresses are extracted. The get_lat_lon function in the Python script serves as an interface with the ArcGIS Geocoding API to convert a physical address into geographical coordinates, specifically latitude and longitude. It begins by preparing a request with the necessary parameters, including the address to be geocoded and the API token for authentication. This request is then sent to the ArcGIS Geocoding API endpoint using the requests.get method. Upon receiving a successful response (indicated by the status code 200), the function parses the JSON data to extract the first candidate's geographical coordinates. If the address is successfully found, the function returns the corresponding longitude and latitude. In cases where the address is not found or if the API request fails, it returns a relevant message indicating the issue. This streamlined function thus encapsulates the entire process of translating an address into its geographical coordinates using the ArcGIS service.

Then a graph is created that contains the origin and destination of the route and connected roads. The graph is shown in **Figure 21**. In order to decrease the computational load network graph of a certain area is extracted rather than loading graph for the entire city or the county. A specific geographic point, defined by latitude and longitude coordinates (33.212917901924726, -87.5424690041741), is set as the focal point for the route creation. Around this point, a network graph was generated using OSMnx, encompassing a 1000-meter radius and considering all types of networks ('all' network_type). The graph visually represents the network of streets and pathways. The route was structured to pass through three key coordinates: a start point, an intermediate point, and an end point, with each pair of points representing distinct segments of the overall route. The intermediate point is used to make sure the shortest route matches with the route that is selected for doing the experiment. Using OSMnx's 'nearest_nodes' function, the nearest network nodes to these coordinates were identified, serving as waypoints for the route. Subsequently, the shortest paths between these nodes were computed using NetworkX's 'shortest_path' function, with the path length ('length') as the weighting factor. This computation was performed separately for the segments from the start to the intermediate point, and from the intermediate to the end point. The two computed paths were then concatenated into a single route, carefully excluding the repeated intermediate node to avoid redundancy. This concatenated path represented the shortest route traversing through the specified waypoints.

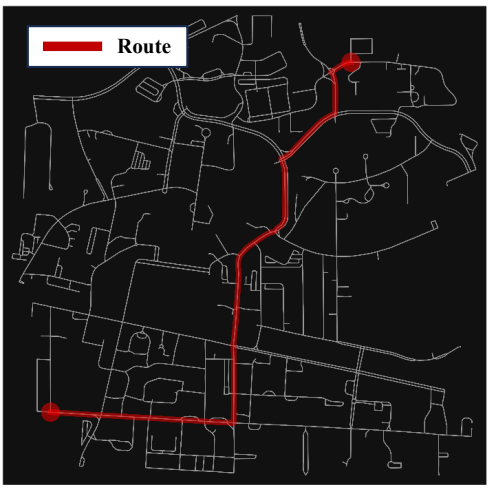


Figure 21: Graph containing roads, intersections, and shortest route between origin and destination.

The subsequent phase of the study involves the extraction and documentation of key features along the route, i.e., the intersections and the entry and exit points of parking lots. This process begins with the extraction of node points from the established route graph, with a particular focus on recording their geographic coordinates (latitude and longitude). Through this approach, all intersection locations within the route are successfully identified and documented. However, it is observed that this methodology did not automatically capture the parking lot entry and exit points. As a result, the geographical data for these specific locations are incorporated manually into the dataset. It is anticipated that, in a future, production-ready version of this method, more comprehensive roadway graphs will be available. Such enhanced graphs are expected to include significant parking lot entry and exit points as distinct nodes, thereby simplifying their extraction. An additional complexity is noted in instances of left-turn maneuvers at intersections. Specifically, the methodology identified three separate nodes for what is functionally a single intersection. This occurrence was attributed to the route intersecting three different links during the left turn. To address this, these three nodes were consolidated into a single node representation in the dataset. Each node was then annotated with relevant information regarding its nature, such as whether it was controlled by a traffic light, a stop sign, or marked an entry or exit point of a parking lot. It is important to note that the experimental route did not feature any unsignalized intersections, and thus no tags were assigned for such scenarios. The culmination of this process is the creation of a comprehensive dataset prior to any trip. This dataset encompassed all the tagged nodes along the route, providing an essential resource for the image-based intersection detection system. The dataset's structure and tagging system ensure that each node is accurately represented and categorized, facilitating efficient and precise image-based recognition and analysis during the experimental trials.

The route is created between an origin and destination. The created route is shown in **Figure 22** as yellow line. Next the efficacy in identifying and extracting location of traffic intersections and big parking lot entries and exits are examined. The developed system successfully detected all intersection points along the route, which are essential for the AV's route correction and are indicated by red plus signs in **Figure 23.** However, the identification of major parking lot entries and exits required manual annotation.

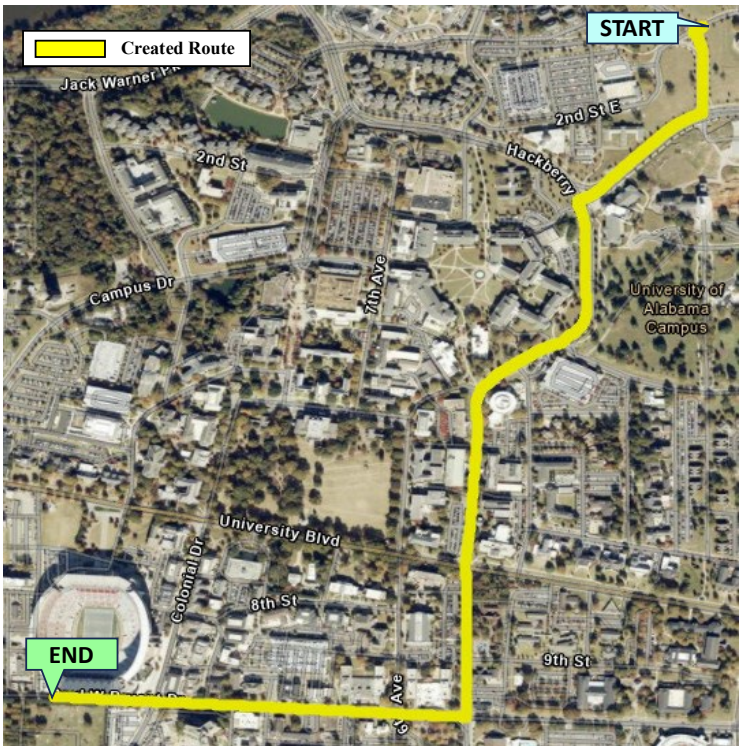


Figure 22: Shortest route between origin (start) and destination (end).

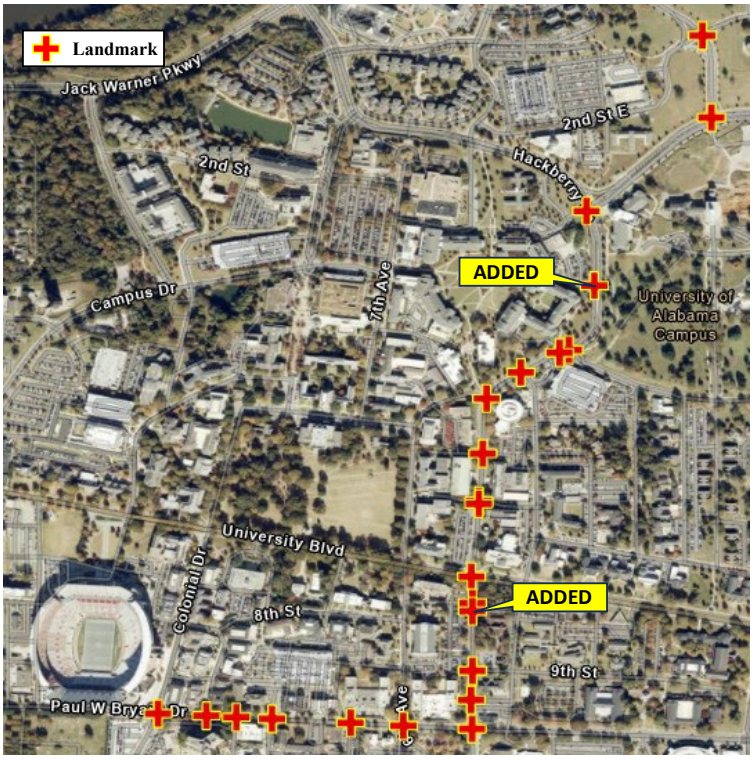


Figure 23: Landmarks along the shortest route

3.3.2.5 Velocity Threshold Determination

The velocity profile presented in **Figure 24** provides a detailed visualization of the vehicle's speed variations over time. The upper portion of the profile illustrates the complete range of velocities encountered during the trip, offering insights into the dynamic behavior of the vehicle, including periods of acceleration, steady movement, and deceleration. The lower, more focused section of the profile zooms in on lower speeds, specifically in the range of 0 to 0.018 meters per second, allowing for a finer examination of the vehicle's behavior at minimal speeds.

Upon cross-referencing this velocity data with camera observations, it was noted that there were instances where the vehicle was stationary, yet the velocity data did not register a velocity of zero. This discrepancy led to a detailed analysis to establish a velocity threshold that could reliably indicate a stopped vehicle in the context of velocity data's inherent noise and accuracy limitations. It was determined that a velocity of less than 0.018 ms^{-1} could be considered as the vehicle being stationary. This threshold was set based on empirical observations that, despite the velocity calculation system's precision, small fluctuations in the calculated position can result in minor speed readings even when the vehicle is not moving. Therefore, setting a velocity threshold of 0.018 meters per second as the criterion for a stopped vehicle accounts for these minor inaccuracies, ensuring that the analysis of vehicle movement remains both accurate and meaningful in real-world conditions.

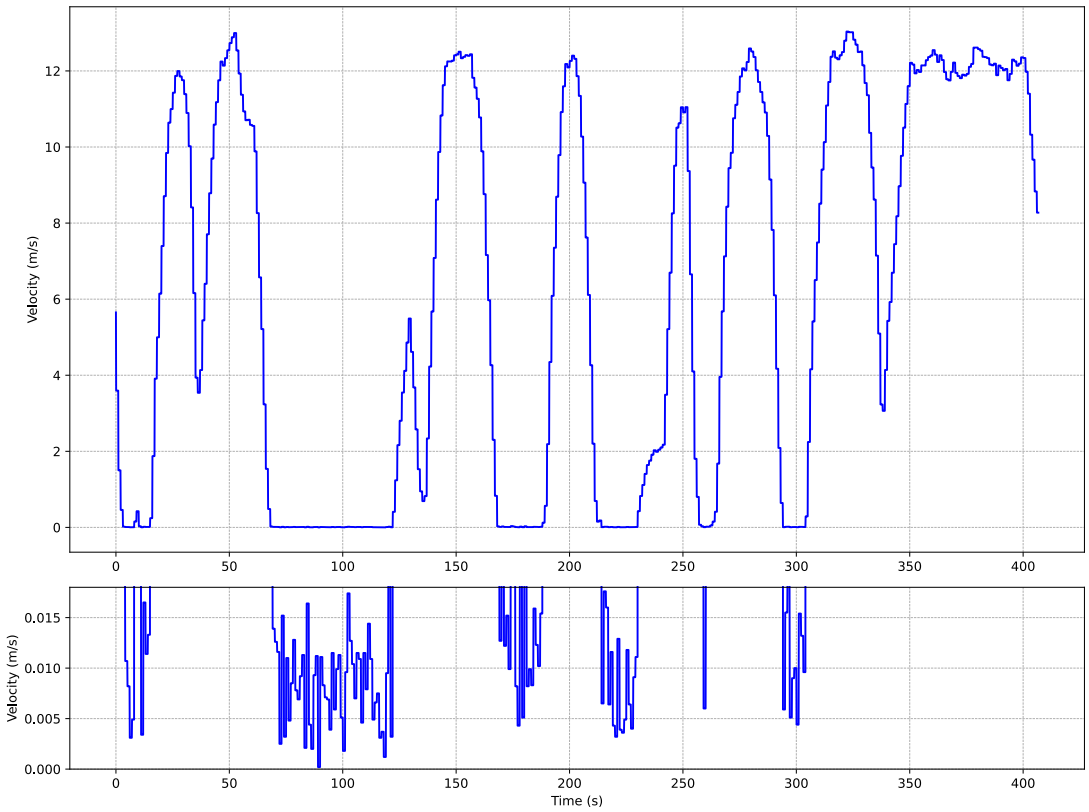


Figure 24: Velocity Profile.

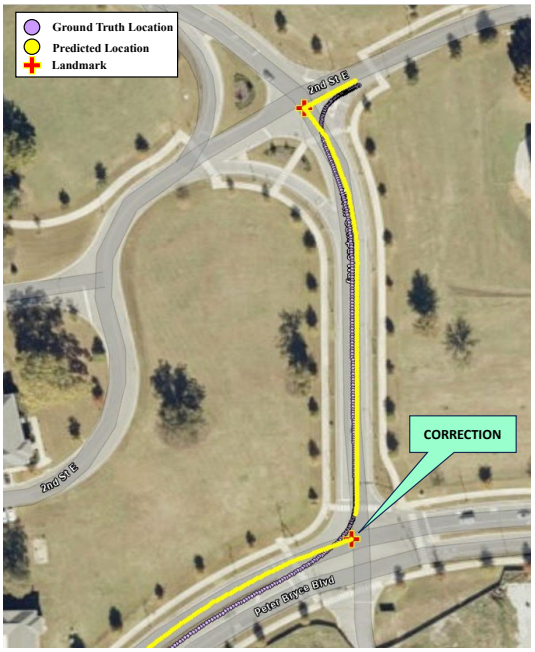


Figure 25: Location correction at a landmark.

3.3.2.6 Location Correction

Figure 25 showcases the effectiveness of the location correction mechanism within our navigation system when it encounters an intersection. Specifically, it demonstrates a scenario where the AV's predicted location, initially at point A, is adjusted to point B to reflect its actual position. This correction is triggered when the AV's onboard computer vision system identifies that the vehicle is at the midpoint of an intersection, rather than the predicted point A. Upon this detection, the system recalibrates the AV's location to point B, which accurately represents where the vehicle truly is at that moment. The process is a crucial part of ensuring the AV's navigation remains reliable and accurate. After the adjustment, the navigation system continues to predict the vehicle's future locations based on this new, corrected position. By doing so, it maintains the integrity of the route guidance and keeps the AV on the correct trajectory, enhancing the overall safety and efficiency of the journey. This dynamic correction capability is essential, especially in complex urban environments where precision in localization can greatly impact navigational decisions and the AV's interaction with its surroundings.

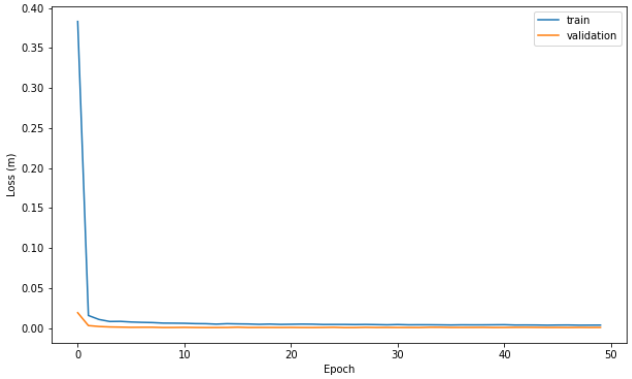


Figure 26: Comparison of mean squared error (loss) profiles with the optimal parameter set.

3.3.2.7 Distance Travelled Prediction (LSTM)

The model achieves a mean absolute error of 0.13m and MSE of 0.026, indication ofa high degree of precision in its predictions. This level of accuracy is a testament to the effectiveness of the chosen hyperparameters and the Adam optimizer's configuration. The model's architecture, coupled with efficient data preprocessing and optimization techniques, demonstrates its capability in accurately predicting the trajectory of the AV, highlighting its potential utility in AV navigation and control systems. **Figure 26** presents the MSE loss profile. Initial epochs exhibit a precipitous decline in both training and validation losses, suggesting rapid learning. As training progresses, the rate of loss reduction abates, eventuating in a plateau. This loss profile is indicative of the model's adeptness in fitting the training dataset while simultaneously demonstrating robustness when exposed to novel data, as inferred from the validation loss trends. Critically, the diminution in loss for both datasets converges to a commensurately low magnitude, indicative of minimal discrepancy between training and validation performance. Such convergence is emblematic of an absence of overfitting, whereby the model eschews memorization of idiosyncrasies within the training data in favor of extracting underlying patterns. Concurrently, the validation loss not only mirrors the decrement observed in the training loss but also stabilizes in unison, which is emblematic of a well-generalized model. This observation is reinforced by the close alignment of the loss curves, particularly in the latter stages of training, where divergence is minimal and does not suggest underfitting—a scenario typically characterized by persistently high training and validation losses. The model's performance trajectory, characterized by convergent and low MSE values for both training and validation datasets, attests to its generalizability and well-calibrated training regimen. Such performance indicates that the model should theoretically maintain efficacy when applied to both the training corpus and previously unseen data. The mean absolute error is subtracted from the predicted distance travelled before feeding to the AV Localization module.

The methods presented in this study offer a structured approach to addressing GNSS spoofing threats in autonomous vehicle navigation. The slow-drift GNSS spoofing attack model effectively simulates real-world adversarial scenarios, allowing for an in-depth analysis of spoofing techniques. The sensor fusion-based spoofing detection framework, leveraging DL and multi-sensor data integration, provides a robust mechanism for identifying and mitigating spoofing attempts. Additionally, the GIS and landmark-based navigation system offers a practical alternative for maintaining localization accuracy in GNSS-contested environments. Together, these methodologies form a comprehensive strategy for improving the security and resilience of AV navigation, ensuring reliable operation under both normal and compromised GNSS conditions.

CHAPTER 4

Results

4.1 Evaluation of Slow-Drift GNSS Spoofing Attacks for Autonomous Vehicles

A particular intersection from the experiment's driving route is designated for in-depth analysis. Two trajectories are singled out for examination: the right turn trajectory is treated as the legitimate route, whereas the left turn trajectory is labeled as the spoofed route (See **Figure 27**). The visible GPS satellites for both routes are G05, G10, G13, G15, G18, G23, G24, and G29. Since the number of visible satellites exceeds the minimum requirement of four GPS satellites, it is suitable for location determination. **Figure 28** presents the legitimate and spoofed pseudorange values for each visible satellite and **Figure 29** shows the difference between the legitimate and spoofed pseudorange. For G10, G18, G23, G24 the legitimate pseudorange is higher than the spoofed, and G05, G13, G15, G29 legitimate pseudorange is lower than the spoofed. Hence, for spoofed pseudorange of all the visible satellites will not be always higher or lower than the legitimate one which makes it hard to predict the spoofed pseudorange and needs multiple iterations to determine the correct set of spoofed pseudoranges for mimicking the spoofed location. We are doing further studies to understand these differences. Based on the experimental data, the correlation between the legitimate and the spoofed pseudorange for each satellite is established and shown in **Figure 30.** The R^2 values vary between 0.99 and 1 showing the response variable (legitimate) can be accurately explained by the predicted variable (spoofed).

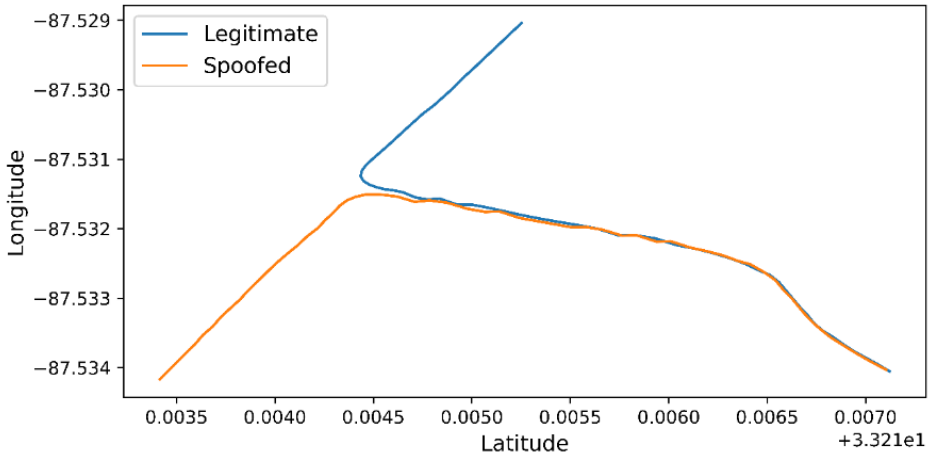


Figure 27: Legitimate and spoofed route.

This outcome shows that it is possible to emulate and/or estimate the desired spoofed pseudorange values by extracting legitimate pseudorange data from the observation data and combining it with the pre-defined spoofed location, which can be fed into the Spoofed Pseudorange Emulator (SPE). Using these calculated spoofed pseudoranges and legitimate CEI (Clock, Ephemeris, and Integrity) variables from the navigation data, one can determine the spoofed GPS receiver location to slowly guide an AV towards a desired location as per spoofer intention. Especially it is true for urban structured road network systems. As the spoofed signal mirrors the satellites visible in the legitimate signal, exhibiting a seamless transition with no abrupt changes in signal or message properties after the attacker locks onto the AV's receiver. This pre-determined frequency of the spoofed location changes can be designed to mimic a gradual drift from the AV's actual route.

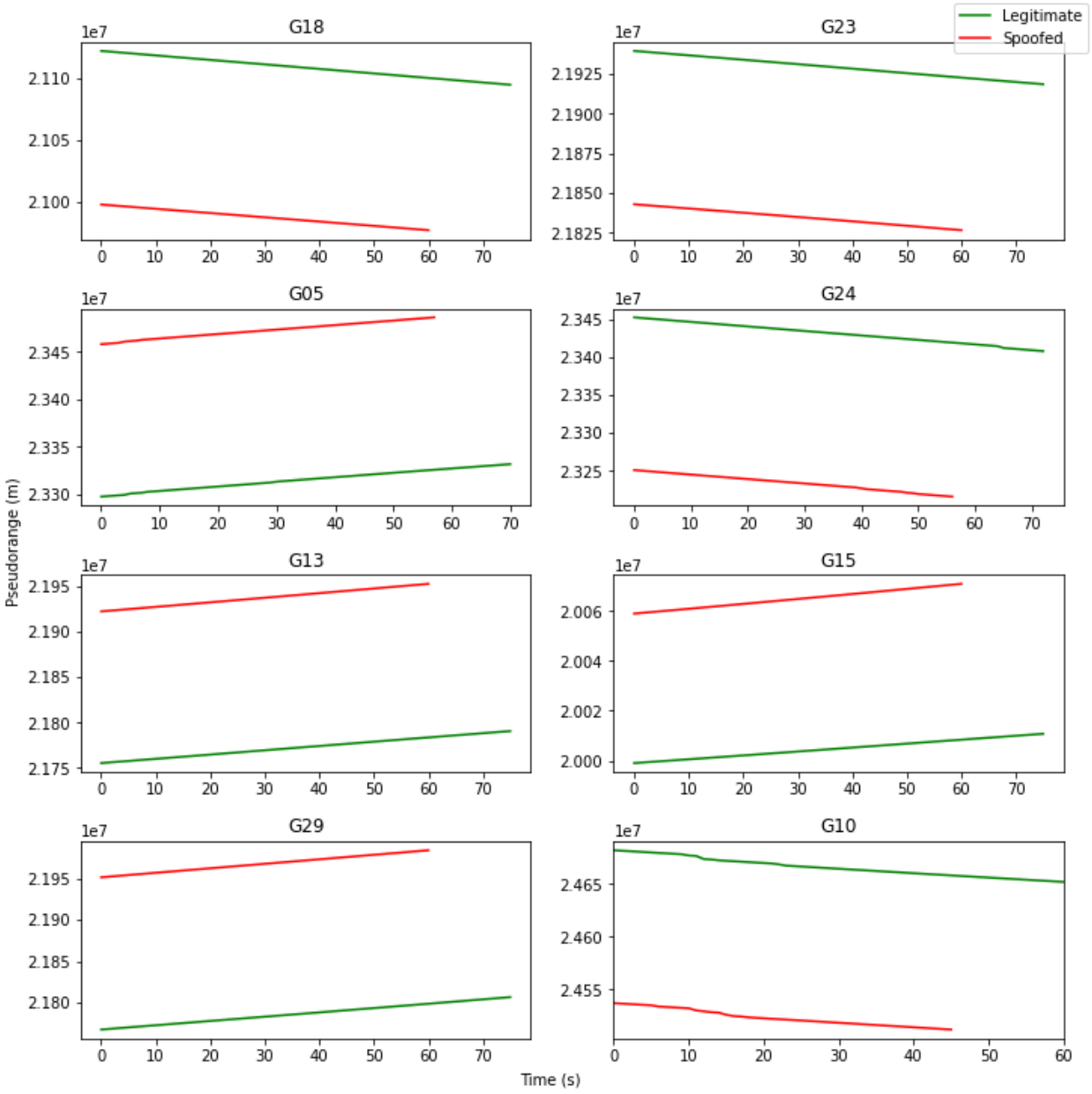


Figure 28: Legitimate and spoofed signal pseudorange.

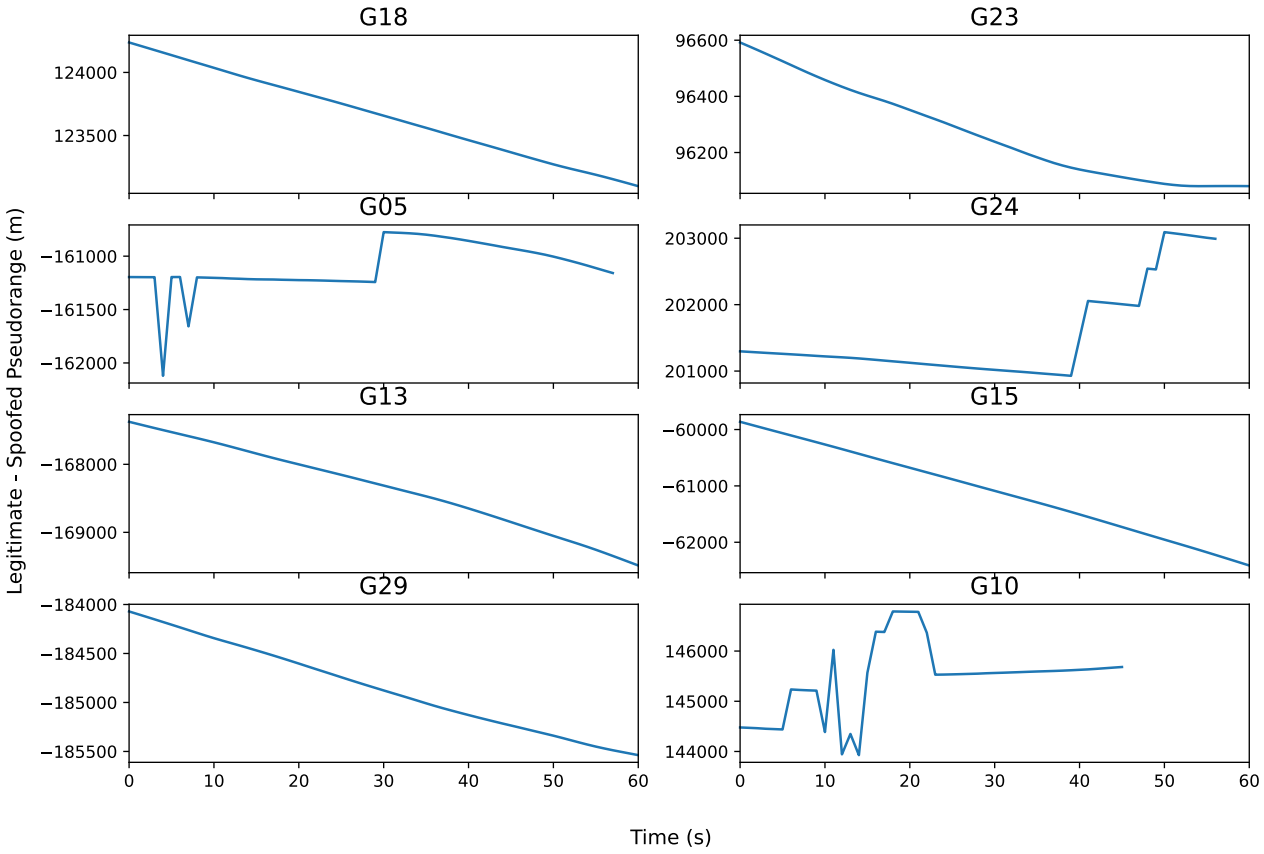
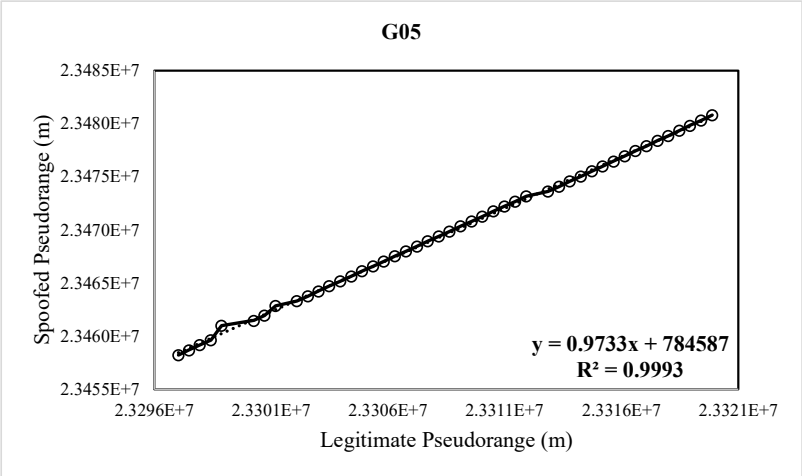
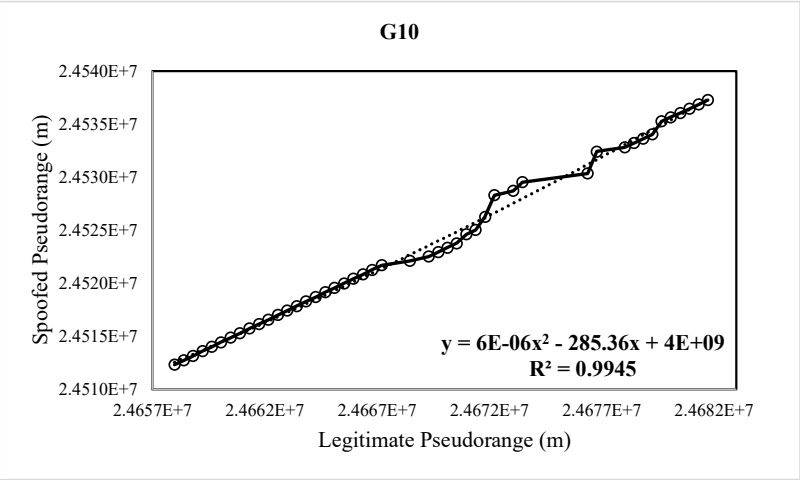


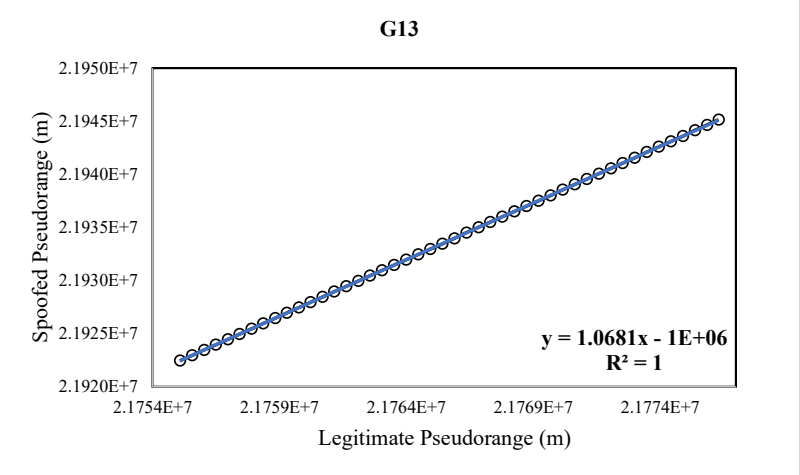
Figure 29: Difference between legitimate and spoofed signal pseudorange.



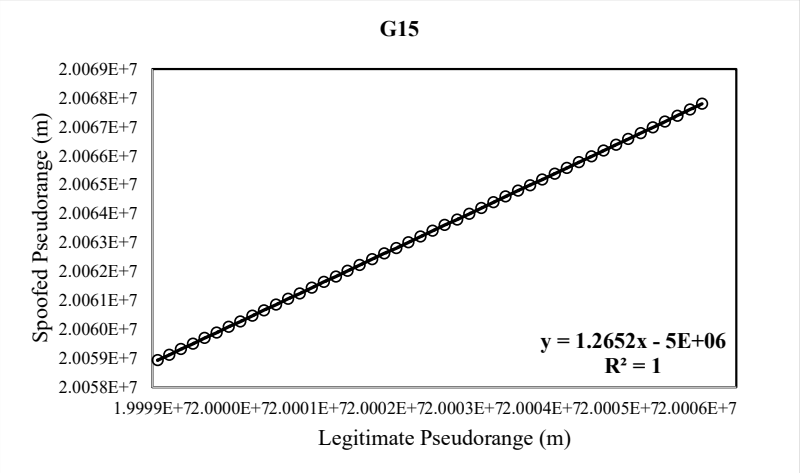
30 (a)


G10
Spoofed Pseudorange (m)
2.4540E+7
2.4535E+7
2.4530E+7
2.4525E+7
2.4520E+7
2.4515E+7
2.4510E+7
y = 6E-06x² - 285.36x + 4E+09
R² = 0.9945
2.4657E+7
2.4662E+7
2.4667E+7
2.4672E+7
2.4677E+7
2.4682E+7
Legitimate Pseudorange (m)

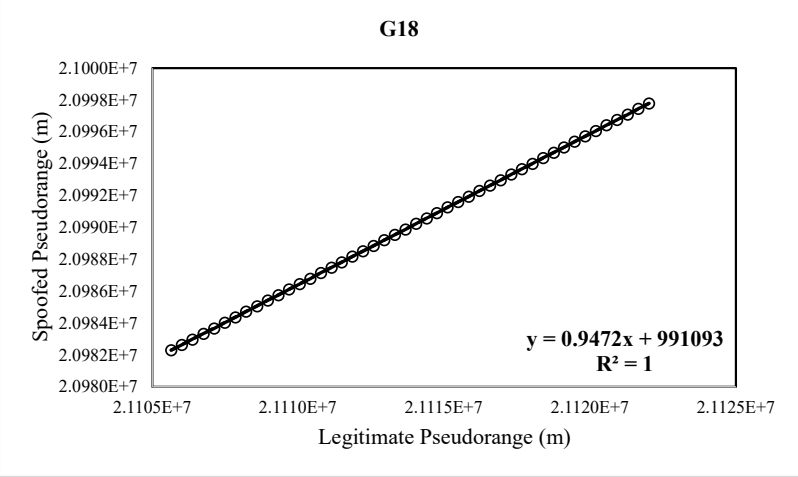
30 (b)


G13
Spoofed Pseudorange (m)
2.1950E+7
2.1945E+7
2.1940E+7
2.1935E+7
2.1930E+7
2.1925E+7
2.1920E+7
y = 1.0681x - 1E+06
R² = 1
2.1754E+7
2.1759E+7
2.1764E+7
2.1769E+7
2.1774E+7
Legitimate Pseudorange (m)

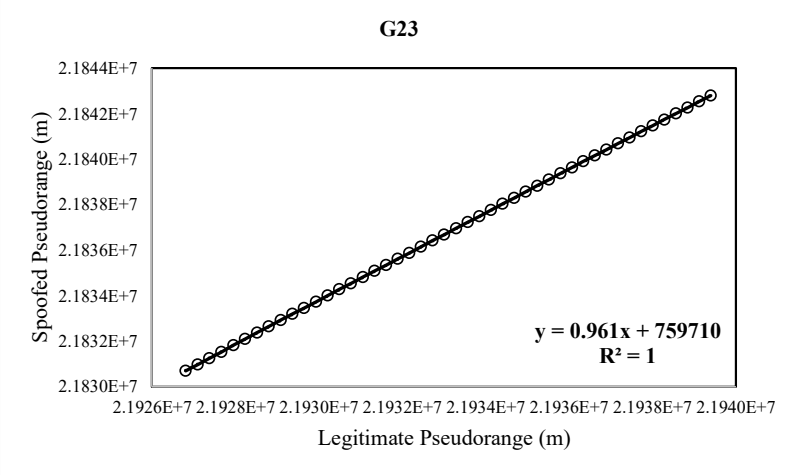
30 (c)


G15
Spoofed Pseudorange (m)
2.0069E+7
2.0068E+7
2.0067E+7
2.0066E+7
2.0065E+7
2.0064E+7
2.0063E+7
2.0062E+7
2.0061E+7
2.0060E+7
2.0059E+7
2.0058E+7
y = 1.2652x - 5E+06
R² = 1
1.9999E+72.0000E+72.0001E+72.0002E+72.0003E+72.0004E+72.0005E+72.0006E+7
Legitimate Pseudorange (m)

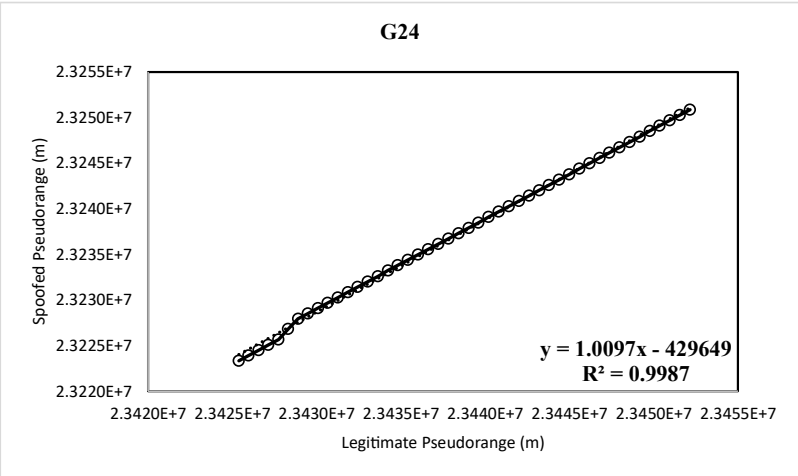
30 (d)


G18
Spoofed Pseudorange (m)
2.1000E+7
2.0998E+7
2.0996E+7
2.0994E+7
2.0992E+7
2.0990E+7
2.0988E+7
2.0986E+7
2.0984E+7
2.0982E+7
2.0980E+7
y = 0.9472x + 991093
R² = 1
2.1105E+7
2.1110E+7
2.1115E+7
2.1120E+7
2.1125E+7
Legitimate Pseudorange (m)

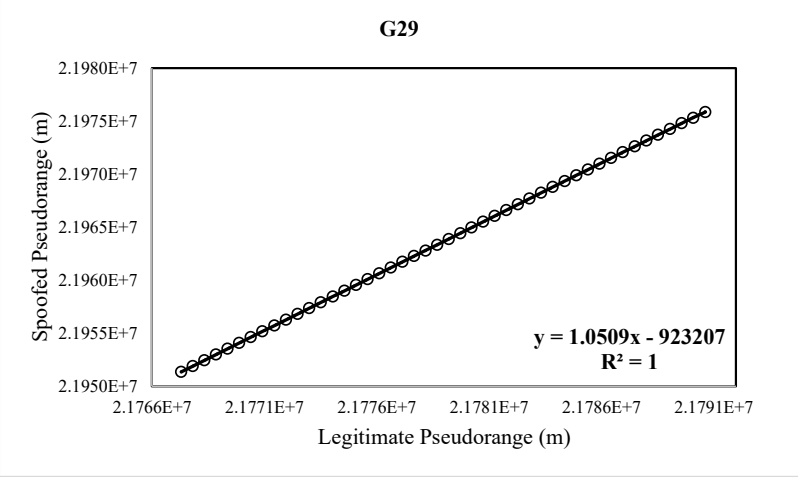
30 (e)


G23
Spoofed Pseudorange (m)
2.1844E+7
2.1842E+7
2.1840E+7
2.1838E+7
2.1836E+7
2.1834E+7
2.1832E+7
2.1830E+7
y = 0.961x + 759710
R² = 1
2.1926E+7 2.1928E+7 2.1930E+7 2.1932E+7 2.1934E+7 2.1936E+7 2.1938E+7 2.1940E+7
Legitimate Pseudorange (m)

30 (f)


G24
Spoofed Pseudorange (m)
2.3255E+7
2.3250E+7
2.3245E+7
2.3240E+7
2.3235E+7
2.3230E+7
2.3225E+7
2.3220E+7
y = 1.0097x - 429649
R² = 0.9987
2.3420E+7 2.3425E+7 2.3430E+7 2.3435E+7 2.3440E+7 2.3445E+7 2.3450E+7 2.3455E+7
Legitimate Pseudorange (m)

30 (g)



30 (h)

Figure 30: Correlation between legitimate and emulated spoofed pseudorange.

4.2 Performance Evaluation of Sensor Fusion-Based GNSS Spoofing Attack Detection Framework

The attack detection framework is evaluated for two distinct attack types: turn-by-turn and wrong turn. Evaluations are conducted in real time by initiating attacks through the injection of spoofed latitude and longitude data at specific points during the trip, accurately mimicking actual GNSS spoofing incidents. The driving route used for evaluation is depicted in **Figure 31**. For turn-by-turn attacks, three instances are conducted along the route, involving shifts to new locations (adjacent blocks) three times, each representing separate events. Additionally, six wrong turn attacks are conducted, where three cases involve spoofed routes indicating left turns while the AV takes right turns, and the other three cases represent spoofed routes indicating right turns while the AV is making left turns in reality. The turn-by-turn attack detection model achieves a perfect detection accuracy, successfully identifying all three location shifts with 100% accuracy.

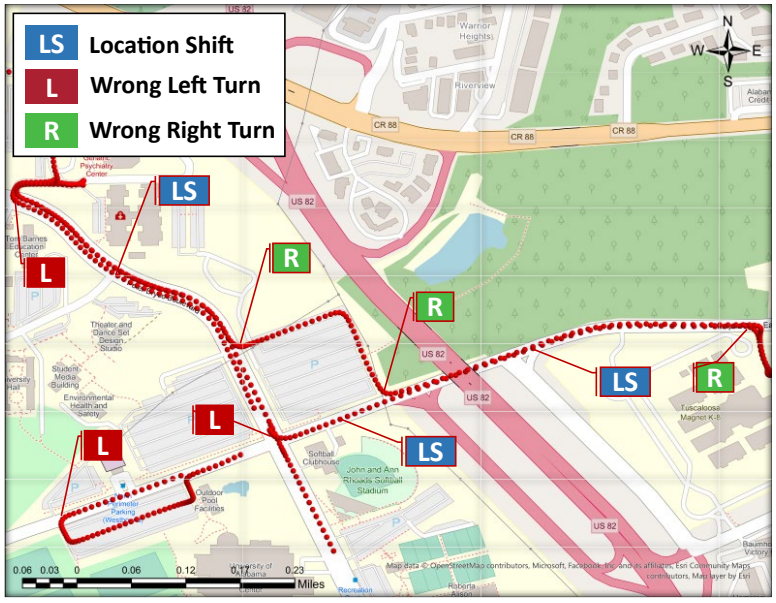


Figure 31: Testing route.

Table 11 presents the results of the wrong turn detection algorithm in terms of precision, recall, accuracy, and F1 score. Precision measures the accuracy of detecting wrong-turn attacks among all the attack detection instances considered in this study. The precision for detecting wrong right turns is 95%, and for wrong left turns, it is 96%. Recall, on the other hand, indicates the percentage of observations where attacks are correctly detected among all the compromised observations. As shown in **Table 11**, the recall for wrong turns is 1, indicating that all wrong turn attacks are successfully identified. The accuracy of detecting wrong right turns is 95%, and for wrong left turns, it is 96%. The F1 score, which balances precision and recall, is 0.97 for wrong right turns and 0.98 for wrong left turns, demonstrating a well-balanced performance. Thus, the attack detection framework excels in real-time detection of wrong turn attacks, whether they involve right turns or left turns.

Table 11: Random Forest model validation result.

Turn type	precision	recall	accuracy	f1-score
Right	0.95	1	0.95	0.97
Left	0.96	1	0.96	0.98

4.3 Evaluation of GIS and Landmark-Based Navigation System for Autonomous Vehicles

The efficacy of a navigation framework for AVs in an urban environment has been rigorously evaluated with an emphasis on real-time vehicle localization and path tracking.

4.3.1 Location Accuracy

Figure 32 illustrates the performance of a localization method using LSTM and KF by plotting the location error over the course of a journey. The error represents the error between the predicted and actual locations of the vehicle, determined using the Haversine formula which accounts for the earth's curvature in calculating distance. The plot's X-axis marks the progression of time through sequential timestamps, while the Y-axis measures the error in meters. Error associated with the LSTM method, is shown in blue, with its mean error indicated by a green dashed line. Error associated with the KF method, is shown in orange, with its mean error indicated by a red dashed line. The plot also includes dotted lines representing the mean error plus and minus three standard deviations (3σ) for each method. These lines indicate the variability of the errors from the mean and give an idea of the consistency of each method. The horizontal purple lines represent the time when a landmark is detected, and location correction is performed. It is evident from the figure that corrections make sure that the location error does not keep increasing. These corrections are integral to the method, enabling the AV to recalibrate its position and successfully navigate towards its intended destination. The LSTM method has a mean error of approximately 6.23m, indicating the average deviation from the true position over the course of the data recorded. Its standard deviation, a measure of error variability, is about 4.15m, which suggests that most of the LSTM error values lie within this range from the mean. A key observation from the analysis is that there are no outliers beyond three standard deviations (3σ) from the mean for the LSTM method, highlighting a degree of stability in its performance. In contrast, the KF method demonstrates a more accurate and consistent performance with a lower mean error of approximately 3.44m. This suggests that on average, the KF method's predictions are closer to the true position than those of the LSTM method. Moreover, the KF method's standard deviation is around 2.13m, which is lower than that of the LSTM method, indicating that its errors are less dispersed and hence more consistent. The analysis shows that there are 15 instances where the KF errors fall outside the ±3σ range, categorizing them as outliers. These deviations imply that while the KF method is generally more accurate and consistent, it is prone to occasional significant errors though the outliers' errors are still less than corresponding LSTM errors. Considering both the mean errors and the standard deviations, the KF

method has a 44.73% improvement in mean error and a 48.52% improvement in standard deviation over the LSTM method. This substantial percentage improvement underscores the enhanced accuracy and consistency of the KF method, making it a significantly better performer in this specific context. Finally, this analysis shows that the localization method is robust, and the implemented correction mechanism is effective in ensuring the AV's navigation remains on course. The ability to self-correct in response to positional errors is a crucial aspect of the AV's navigation system, as it ensures high accuracy and enhances the safety and reliability of the vehicle's autonomous operations.

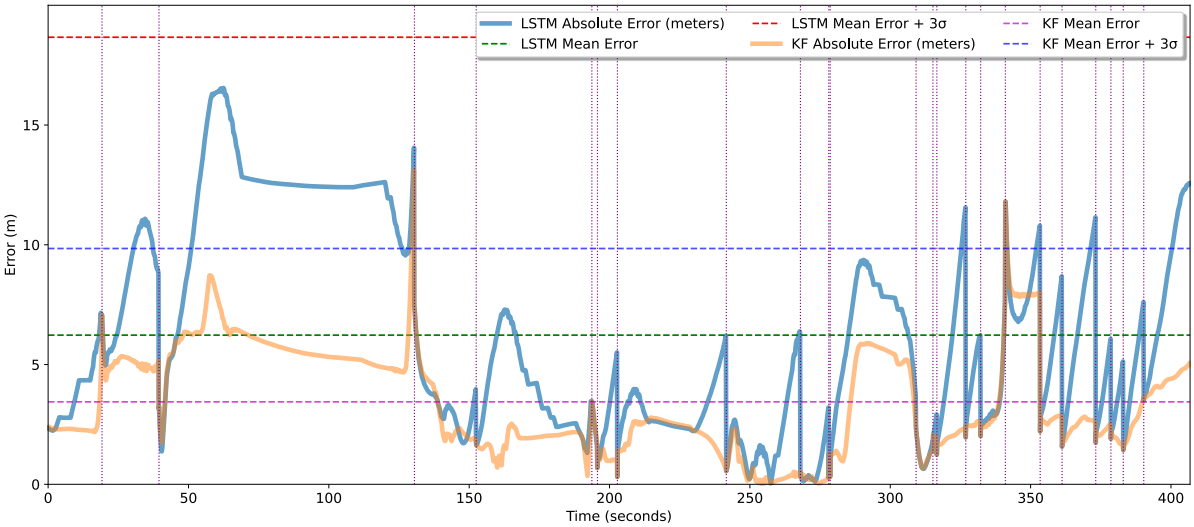


Figure 32: Location error profile along route.

Figure 33 represents a histogram that details the frequency distribution of positional errors. Each bar in the histogram corresponds to a range of error values, measured in meters, with the height of the bar indicating how frequently each range of errors was observed during the evaluation. The X-axis of the histogram represents the error in meters, segmented into bins that group the error values into specific ranges. The Y-axis indicates the frequency of each error range occurring. The height of each bar corresponds to the number of occurrences of the error values within that particular bin.

The LSTM histogram of positional errors reveals a bi-modal distribution, indicating the presence of two distinct peaks. These peaks suggest that there are two prevalent error magnitudes within the dataset. The first peak, closer to the mean, represents a larger concentration of data points indicating more frequent, smaller errors. This peak is where the majority of the data is concentrated, as shown by the histogram and the standard deviation lines, with 66.2% of the data within ±1σ of the mean. The second peak, located further to the right, indicates another common error magnitude but with less frequency compared to the first. The presence of this second peak suggests that there is a secondary pattern of error, potentially attributable to a different cause or set of conditions than those contributing to the primary error cluster. This bi-modality is atypical for normally distributed data, which would typically have a single central peak. The rightward skewness value of approximately 0.60 is consistent with the visual representation, where the tail extends more prominently towards the right, indicating that larger errors occur less frequently but are still significant. The platykurtic nature of the distribution (kurtosis of approximately -0.82) implies that the distribution is relatively spread out with fewer outliers than a normal distribution, which can be Seen in the flattened, wider spread of the histogram's tails. In terms of the data's spread around the mean, about 66.2% of the errors fall within one standard deviation (±1σ), around 97.4% within two standard deviations (±2σ), and the entirety of the dataset falls within three standard deviations (±3σ) of the mean. This indicates that

while there is some variability, the majority of errors are concentrated around the mean and within a relatively predictable range, with very few extreme values.

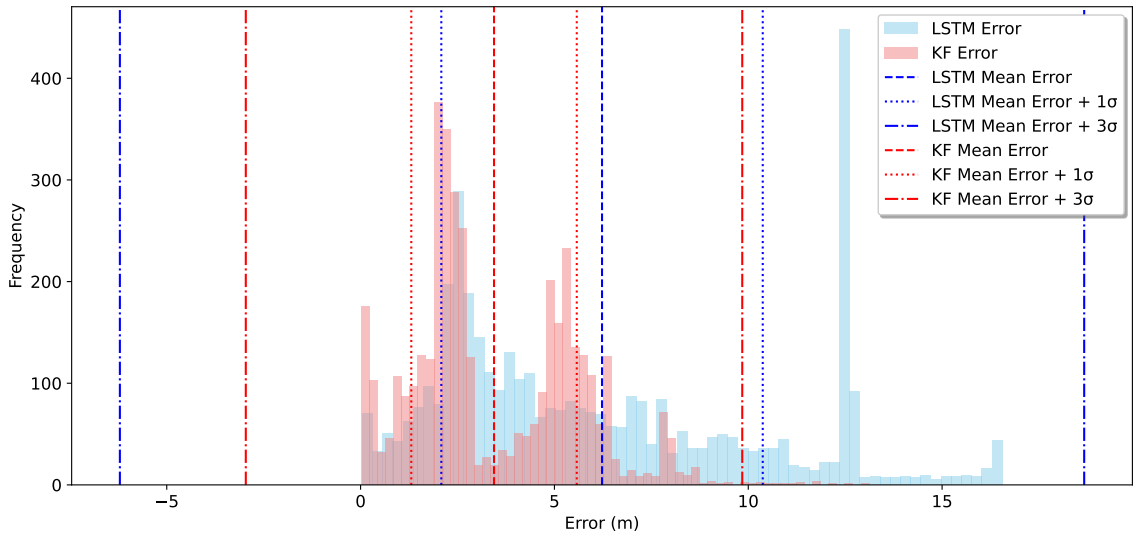


Figure 33: Location error distribution.

The error distribution for the KF method appears predominantly unimodal rather than bimodal, concentrating most data points near the mean error. The distribution has a rightward skewness with a value of approximately 0.55, suggesting a longer tail towards the larger error values, indicating that while less frequent, significant errors do occur. The kurtosis value of approximately -0.22 points to a platykurtic distribution, which is relatively flatter than a normal distribution, implying a broader spread of the error values and fewer outliers. In terms of dispersion around the mean error, about 68.73% of the error values lie within one standard deviation (±1σ), around 95.60% within two standard deviations (±2σ), and approximately 99.63% within three standard deviations (±3σ) from the mean. This indicates a high level of predictability in the error distribution, with the vast majority of errors falling close to the mean, demonstrating the method's consistency. There are very few extreme deviations, as evidenced by the small percentage of errors falling outside the ±3σ range. Overall, the KF method's error distribution suggests that it is both accurate, given the low mean error, and reliable, considering the high percentage of errors within one standard deviation of the mean. The histogram affirms the quantitative analysis, further solidifying the KF method's efficacy in providing precise localization with a predictable and consistent error range.

4.3.2 Navigation Along the Route

Using a combination of GNSS-derived velocity (in place of speedometer data), IMU measurements, and predictions from an LSTM model, the navigation system is designed to function reliably even when GNSS signals are unreliable or unavailable. **Figure 34** and **Figure 35** illustrates both the predicted and ground-truth locations of the vehicle predicted by LSTM and KF method respectively. The close alignment of the predicted path with the actual recorded path, as Seen in the plot, confirms the system's capability to accurately navigate the AV to its destination, independent of GNSS input.

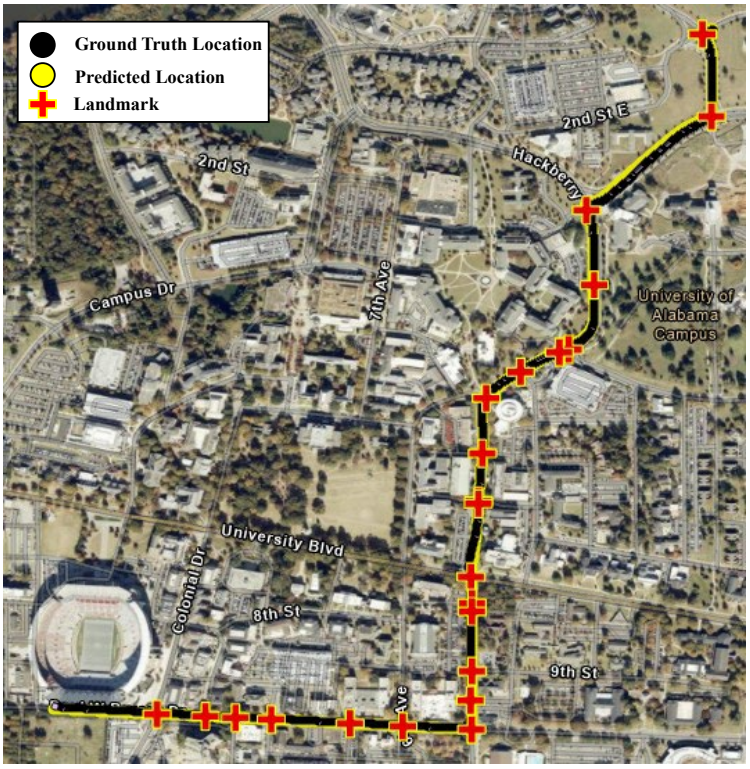


Figure 34: Ground-truth and predicted location (LSTM)

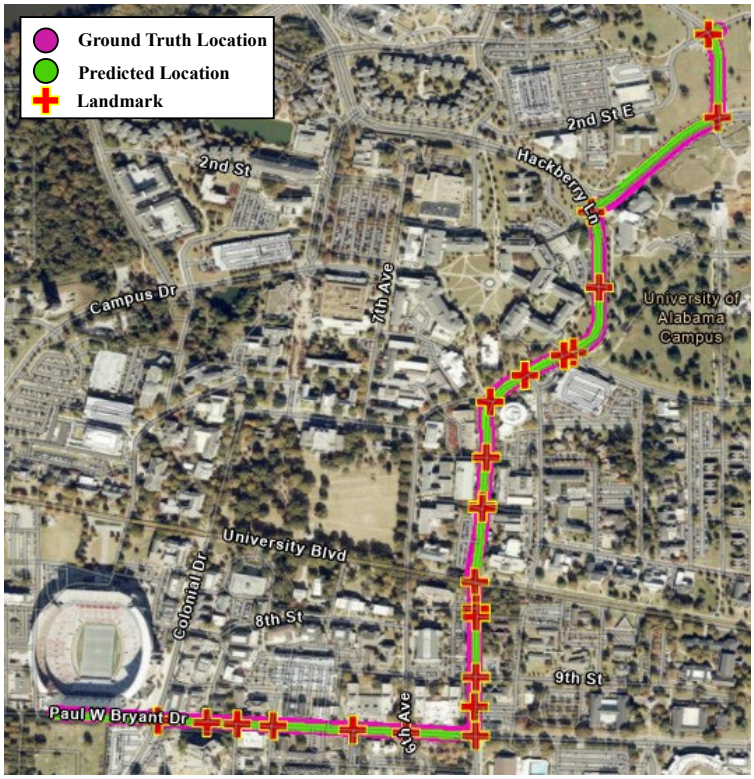


Figure 35: Ground-truth and predicted location (KF).

The results of this study validate the effectiveness of the proposed approaches in mitigating GNSS spoofing threats. The slow-drift GNSS spoofing attack successfully demonstrated how an AV's position can be subtly manipulated, highlighting the need for advanced detection mechanisms. The sensor fusion-based detection framework achieved high accuracy in identifying spoofing-induced anomalies, confirming its potential for real-time implementation in AV systems. Furthermore, the GIS and landmark-based navigation system proved to be a reliable alternative in GNSS-denied environments, significantly improving localization accuracy. These findings collectively reinforce the importance of multi-sensor fusion, machine learning-based detection, and alternative navigation strategies in enhancing the security and operational resilience of autonomous vehicles.

CHAPTER 5

Conclusions

This report provides a comprehensive study of the vulnerabilities of GNSS-based navigation in AVs and presents an integrated approach for detecting and mitigating GNSS spoofing attacks. The study successfully demonstrated that slow-drift GNSS spoofing attacks pose a significant risk to AV operations by gradually altering the vehicle's position over time, even during turns, without immediate detection. The proposed attack method precisely simulated satellite reception patterns and manipulated pseudoranges to create a seamless but misleading navigation experience for the AV. Field experiments confirmed that understanding the relationship between authentic and spoofed pseudoranges is essential for developing effective countermeasures against such sophisticated threats.

To address this challenge, a sensor fusion-based GNSS spoofing attack detection framework was experimentally validated. The framework integrated speedometer data with high-frequency IMU sensor readings to detect inconsistencies between the AV's expected and actual motion. Two complementary detection strategies were implemented: (1) An LSTM neural network for predicting the AV's movement based on in-vehicle sensor data and comparing it with GNSS-derived distance estimates, and (2) A RF machine learning model for turn classification using gyroscope data. Experimental validation confirmed the robustness of this approach, with high accuracy in detecting spoofing attacks. The LSTM-based prediction model effectively identified spoofing-induced anomalies in vehicle motion, while the RF-based turn detection model reliably classified turning maneuvers and detected discrepancies in GNSS data. These results highlight the effectiveness of sensor fusion in detecting GNSS spoofing attacks, demonstrating its potential to enhance AV cybersecurity and navigation integrity.

Additionally, this study introduced a GIS and landmark-based navigation system as a resilient alternative to GNSS in contested environments. The framework integrated IMU data, speedometer readings, and real-time intersection detection through computer vision to maintain accurate localization in GNSS-denied scenarios. A Kalman Filter (KF) was employed for position correction, while an LSTM-based predictive model estimated travel distances. The system was tested in urban environments, where GNSS signals were intentionally degraded. The results demonstrated that landmark-aided navigation significantly improved localization accuracy, effectively enabling AVs to operate without relying on GNSS. The Kalman Filter approach outperformed the LSTM model in distance prediction, demonstrating lower mean errors and more precise localization. These findings confirm the feasibility of GIS and landmark-based navigation as a backup system for AVs when GNSS reliability is compromised.

While this study provides a strong foundation for securing AV navigation against GNSS spoofing threats, several areas warrant further research. Future work should focus on real-time adaptation of spoofing detection algorithms, ensuring that machine learning models can dynamically adjust to evolving attack techniques. Expanding the dataset to include a wider range of spoofing scenarios and varied driving conditions will further validate the robustness of the detection framework. Future studies should also evaluate the framework in more challenging environments, such as dense urban corridors and adverse weather, and examine coordinated multi-vector attacks involving GNSS and V2X systems. In addition, the dependence of the GIS and landmark-based backup navigation framework on detailed maps and visually recognizable roadway features should be further investigated, especially in rural or rapidly changing environments. Incorporating additional sensing modalities, such as LiDAR-based landmark detection, radar-based motion tracking, and ultra-wideband (UWB) localization, could further improve spoofing detection accuracy and enhance navigation resilience. Another key challenge is distinguishing between intentional GNSS spoofing attacks and unintentional GNSS signal disruptions caused by environmental factors. Future research should explore cryptographic GNSS authentication methods, cooperative multi-

vehicle localization, and AI-driven anomaly detection to enhance the reliability of AV navigation systems. Further advancements in GIS and landmark-based navigation could improve AV localization accuracy in diverse environments. Future studies should investigate dynamic landmark recognition using DL-based feature extraction, as well as testing the framework's adaptability in rural roads, tunnels, and off-road environments. Additionally, integrating machine learning for automated feature recognition and real-time route optimization could enhance the system's efficiency and robustness in real-world driving conditions.

Finally, continued collaboration between researchers, industry stakeholders, and government agencies will be crucial in establishing standardized GNSS cybersecurity protocols for AVs. This study demonstrates that multi-sensor fusion techniques, alternative navigation strategies, and machine learning-based detection models play a critical role in securing AV navigation against GNSS spoofing threats. By addressing these challenges, future research can contribute to the development of a cyber-resilient AV ecosystem, ensuring the safe and reliable deployment of autonomous transportation systems in both GNSS-reliable and GNSS-contested environments.

REFERENCES

Abdallah, A.A. (2023) *Cellular Signals for Navigation: 4G, 5G, and Beyond*. UC Irvine.

Abdallah, A.A., Khalife, J. and Kassas, Z.M. (2023) 'Exploiting on-demand 5G downlink signals for opportunistic navigation', *IEEE Signal Processing Letters* [Preprint].

Aldibaja, M., Suganuma, N. and Yoneda, K. (2017) 'LIDAR-data accumulation strategy to generate high definition maps for autonomous vehicles', in *2017 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, pp. 422–428.

Almutairy, F. *et al.* (2023) 'Detection and mitigation of GPS Spoofing Attacks on Phasor Measurement Units using deep learning', *International Journal of Electrical Power & Energy Systems*, 151, p. 109160. Available at: https://doi.org/10.1016/J.IJEPES.2023.109160.

Altaweel, A., Mukkath, H. and Kamel, I. (2023) 'GPS Spoofing Attacks in FANETs: A Systematic Literature Review', *IEEE Access*, 11, pp. 55233–55280. Available at: https://doi.org/10.1109/ACCESS.2023.3281731.

Anderson, P. *et al.* (2018) 'On Evaluation of Embodied Navigation Agents'.

Ang, K.H., Chong, G. and Li, Y. (2005) 'PID control system analysis, design, and technology', *IEEE Transactions on Control Systems Technology*, 13(4), pp. 559–576. Available at: https://doi.org/10.1109/TCST.2005.847331.

Arandjelović, R. *et al.* (2018) 'NetVLAD: CNN Architecture for Weakly Supervised Place Recognition', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(6), pp. 1437–1451. Available at: https://doi.org/10.1109/TPAMI.2017.2711011.

Baldysz, Z. *et al.* (2023) 'Diurnal variability of atmospheric water vapour, precipitation and cloud top temperature across the global tropics derived from satellite observations and GNSS technique', *Climate Dynamics*, 62(3), pp. 1965–1982. Available at: https://doi.org/10.1007/S00382-023-07005-0/TABLES/2.

Bauer, S., Alkhorshid, Y. and Wanielik, G. (2016) 'Using high-definition maps for precise urban vehicle localization', in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 492–497.

Beinhofer, M., Kretzschmar, H. and Burgard, W. (2013) 'Deploying artificial landmarks to foster data association in simultaneous localization and mapping', in *2013 IEEE International Conference on Robotics and Automation*, pp. 5235–5240.

Boehm, J., Werl, B. and Schuh, H. (2006) 'Troposphere mapping functions for GPS and very long baseline interferometry from European Centre for Medium-Range Weather Forecasts operational analysis data', *Journal of Geophysical Research: Solid Earth*, 111(B2), p. 2406. Available at: https://doi.org/10.1029/2005JB003629.

Borhani-Darian, P. *et al.* (2020) 'Deep neural network approach to detect GNSS spoofing attacks', in *Proceedings of the 33rd International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS+ 2020)*, pp. 3241–3252.

Brachmann, E. *et al.* (2021) 'On the Limits of Pseudo Ground Truth in Visual Camera Re-localisation'.

Brosh, E. *et al.* (2021) 'Accurate Visual Localization for Automotive Applications'.

Broumadan, A., Kennedy, S. and Schleppe, J. (2020) 'Nobody's Fool: Spoofing Detection in a High-Precision Receiver', *Inside GNSS* [Preprint]. Available at: https://insidegnss.com/nobodys-fool-spoofing-detection-in-a-high-precision-receiver/.

Buesnel, G. (2020) 'With GNSS Spoofing Attacks on the Rise, Resilience and Robustness Go Hand-in-Hand'.

Crosara, L. *et al.* (2024) 'Worst-Case Spoofing Attack and Robust Countermeasure in Satellite Navigation Systems', *IEEE Transactions on Information Forensics and Security*, 19, pp. 2039–2050. Available at: https://doi.org/10.1109/TIFS.2023.3340061.

Daneshmand, S. *et al.* (2012) 'A Low-Complexity GPS Anti-Spoofing Method Using a Multi-Antenna Array', pp. 1233–1243. Available at: http://www.ion.org/publications/abstract.cfm?jp=p&articleID=10336 (Accessed: 14 June 2021).

D'Antonio, S. *et al.* (2011) 'Security issues of a phasor data concentrator for smart grid infrastructure', *ACM International Conference Proceeding Series*, pp. 3–8. Available at: https://doi.org/10.1145/1978582.1978584.

Dasgupta, S. *et al.* (2022a) 'A Sensor Fusion-Based GNSS Spoofing Attack Detection Framework for Autonomous Vehicles', *IEEE Transactions on Intelligent Transportation Systems* [Preprint]. Available at: https://doi.org/10.1109/TITS.2022.3197817.

Dasgupta, S. *et al.* (2022b) 'A Sensor Fusion-Based GNSS Spoofing Attack Detection Framework for Autonomous Vehicles', *IEEE Transactions on Intelligent Transportation Systems*, 23(12), pp. 23559–23572. Available at: https://doi.org/10.1109/TITS.2022.3197817.

Dasgupta, S. (2024) *Cyber-Resilient Positioning and Navigation for Autonomous Ground Vehicles*, *ProQuest Dissertations and Theses*. Available at: https://www.proquest.com/dissertations-theses/cyber-resilient-positioning-navigation-autonomous/docview/3066667759/se-2.

Dasgupta, S. *et al.* (2024) 'Unveiling the Stealthy Threat: Analyzing Slow Drift GPS Spoofing Attacks for Autonomous Vehicles in Urban Environments and Enabling the Resilience'. Available at: https://arxiv.org/abs/2401.01394v1 (Accessed: 8 February 2025).

Dasgupta, S., Shakib, K.H. and Rahman, M. (2024) 'Experimental Validation of Sensor Fusion-based GNSS Spoofing Attack Detection Framework for Autonomous Vehicles'. Available at: https://arxiv.org/abs/2401.01304v1 (Accessed: 8 February 2025).

Ding, M., Chen, W. and Ding, W. (2023) 'Performance analysis of a normal GNSS receiver model under different types of jamming signals', *Measurement*, 214, p. 112786. Available at: https://doi.org/10.1016/J.MEASUREMENT.2023.112786.

Dooley, C. (2018) 'HD Maps: The key to autonomous driving success'.

Dries, R.F., Pratt, M. and Johnson, R. (2021) 'Methods for Detecting Replay Attacks in GNSS Systems and Devices Thereof'.

Elghazaly, G. *et al.* (2023) 'High-Definition Maps: Comprehensive Survey, Challenges and Future Perspectives', *IEEE Open Journal of Intelligent Transportation Systems* [Preprint]. Available at: https://doi.org/10.1109/OJITS.2023.3295502.

Federal Register :: Strengthening National Resilience Through Responsible Use of Positioning, Navigation, and Timing Services (no date). Available at: https://www.federalregister.gov/documents/2020/02/18/2020-03337/strengthening-national-resilience-through-responsible-use-of-positioning-navigation-and-timing (Accessed: 21 July 2023).

Geomate (2022) 'HD Maps for Autonomous Driving'.

'GPS L1 C/A Receiver Processing' (2022) in *GNSS Software Receivers*. Cambridge University Press, pp. 108–125.

Gurtner, W. (2018) 'RINEX The Receiver Independent Exchange Format Version 3.04 International GNSS Service (IGS), RINEX Working Group and Radio Technical Commission for Maritime Services Special Committee 104 (RTCM-SC104)'.

Hata, A. and Wolf, D. (2014) 'Road marking detection using LIDAR reflective intensity data and its application to vehicle localization', in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*, pp. 584–589.

Hata, A.Y. and Wolf, D.F. (2015) 'Feature detection for vehicle localization in urban environments using a multilayer LIDAR', *IEEE Transactions on Intelligent Transportation Systems*, 17(2), pp. 420–429.

Hess, W. *et al.* (2016) 'Real-time loop closure in 2D LIDAR SLAM', in *2016 IEEE international conference on robotics and automation (ICRA)*, pp. 1271–1278.

Hevo Data (2024) 'MongoDB vs SQL Server: Understanding the Differences'.

Hofmann-Wellenhof, B., Lichtenegger, H. and Collins, J. (2012) *Global positioning system: theory and practice*. Available at: https://books.google.com/books?hl=en&lr=&id=F7jrCAAAQBAJ&oi=fnd&pg=PR19&ots=zYkduXvLlw&sig=91HA7AD7VmUWC5HeCRebjW1cFl8 (Accessed: 3 March 2024).

Hu, Y. *et al.* (2018) 'A Novel Array-Based Spoofing and Jamming Suppression Method for GNSS Receiver', *IEEE Sensors Journal*, 18(7), pp. 2952–2958. Available at: https://doi.org/10.1109/JSEN.2018.2797309.

Humphreys, T.E. *et al.* (2008) 'Assessing the Spoofing Threat: Development of a Portable GPS Civilian Spoofer', in *Proceedings of the 21st International Technical Meeting of the Satellite Division of The Institute of Navigation (ION GNSS 2008)*. Savannah, GA, pp. 2314–2325. Available at: http://www.ion.org/publications/abstract.cfm?jp=p&articleID=8132 (Accessed: 28 July 2023).

Ilci, V. and Toth, C. (2020) 'High definition 3D map creation using GNSS/IMU/LiDAR sensor integration to support autonomous vehicle navigation', *Sensors*, 20(3), p. 899.

Kassas, Z.M. and Abdallah, A. (2023) 'No GPS No Problem: Exploiting Cellular OFDM-Based Signals for Accurate Navigation', *IEEE Transactions on Aerospace and Electronic Systems* [Preprint].

Kassas, Z.M., Khairallah, N. and Kozhaya, S. (2024) 'Ad astra: Simultaneous tracking and navigation with megaconstellation LEO satellites', *IEEE Aerospace and Electronic Systems Magazine* [Preprint].

Khalife, J. and Kassas, Z.M. (2023) 'Performance-driven design of carrier phase differential navigation frameworks with megaconstellation LEO satellites', *IEEE Transactions on Aerospace and Electronic Systems* [Preprint].

Khan, Z. *et al.* (2020) 'Long Short-Term Memory Neural Network-Based Attack Detection Model for In-Vehicle Network Security', *IEEE Sensors Letters*, 4(6). Available at: https://doi.org/10.1109/LSENS.2020.2993522.

Kleiner, A., Prediger, J. and Nebel, B. (2006) 'RFID technology-based exploration and SLAM for search and rescue', in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4054–4059.

Křehl\'\ik, Š., Vanžura, M. and Skokan, A. (2023) 'Minimum required accuracy for HD maps', *The Journal of Navigation*, 76(2–3), pp. 238–254.

Kumar, S. *et al.* (2023) 'Study of the atmospheric and ionospheric phenomenon using GPS-based remote sensing technique', *Atmospheric Remote Sensing: Principles and Applications*, pp. 261–282. Available at: https://doi.org/10.1016/B978-0-323-99262-6.00019-5.

Lagler, K. *et al.* (2013) 'GPT2: Empirical slant delay model for radio space geodetic techniques', *Wiley Online LibraryK Lagler, M Schindelegger, J Böhm, H Krásná, T NilssonGeophysical research letters, 2013•Wiley Online Library*, 40(6), pp. 1069–1073. Available at: https://doi.org/10.1002/grl.50288.

Landskron, D. and Böhm, J. (2018) 'VMF3/GPT3: refined discrete and empirical troposphere mapping functions', *Journal of Geodesy*, 92(4), pp. 349–360. Available at: https://doi.org/10.1007/S00190-017-1066-2/TABLES/8.

Lenhart, M., Spanghero, M. and Papadimitratos, P. (2021) 'Relay/replay attacks on GNSS signals', *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pp. 380–382. Available at: https://doi.org/10.1145/3448300.3468256.

Levy, B. and Stern, A. (2019) 'Method and Apparatus for Generating a Precise Time Signal in a Satellite Navigation Receiver'.

Li, Q. *et al.* (2020) 'Multi-sensor fusion for navigation and mapping in autonomous vehicles: Accurate localization in urban environments', *Unmanned Systems*, 8(03), pp. 229–237.

Liu, M. *et al.* (2024) 'A Fast and Accurate Initialization Method for Mocap-Inertial Navigation System', *IEEE Sensors Journal*, p. 1. Available at: https://doi.org/10.1109/JSEN.2024.3357864.

Liu, S. *et al.* (2021) *Stars Can Tell: A Robust Method to Defend against {GPS} Spoofing Attacks using Off-the-shelf Chipset*, *30th {USENIX} Security Symposium ({USENIX} Security 21)*. {USENIX} Association. Available at: https://www.usenix.org/conference/usenixsecurity21/presentation/ma (Accessed: 27 January 2022).

Liu, X. *et al.* (2023) 'Mitigating GNSS multipath in landslide areas: A novel approach considering mutation points at different stages', *Landslides*, 20(11), pp. 2497–2510. Available at: https://doi.org/10.1007/S10346-023-02117-4/FIGURES/12.

Manickam, S. and O'Keefe, K. (2016) 'Using Tactical and MEMS Grade INS to Protect Against GNSS Spoofing in Automotive Applications', *Proceedings of ION GNSS+2016* [Preprint].

Van Der Merwe, J.R. *et al.* (2018) 'Classification of Spoofing Attack Types', *2018 European Navigation Conference (ENC)*, pp. 91–99. Available at: https://doi.org/10.1109/EURONAV.2018.8433227.

Milz, S. *et al.* (2018) 'Visual slam for automated driving: Exploring the applications of deep learning', in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 247–257.

Montenbruck, O., Steigenberger, P. and Hauschild, A. (2020) 'Comparing the "Big 4" - A User's View on GNSS Performance', *2020 IEEE/ION Position, Location and Navigation Symposium, PLANS 2020*, pp. 407–418. Available at: https://doi.org/10.1109/PLANS46316.2020.9110208.

Motallebighomi, M. *et al.* (2023) 'Location-independent GNSS Relay Attacks: A Lazy Attacker's Guide to Bypassing Navigation Message Authentication', *Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, pp. 365–376. Available at: https://doi.org/10.1145/3558482.3590186.

Naoki Akai, L.Y.M. and Murase, H. (2018) 'Simultaneous pose and reliability estimation using convolutional neural network and Rao–Blackwellized particle filter', *Advanced Robotics*, 32(17), pp. 930–944. Available at: https://doi.org/10.1080/01691864.2018.1509726.

Neish, A. *et al.* (2018) 'Uncoupled accelerometer based GNSS spoof detection for automobiles using statistic and wavelet based tests', in *Proceedings of the 31st International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS+ 2018*. Institute of Navigation, pp. 2938–2962. Available at: https://doi.org/10.33012/2018.15903.

Odukha, O. (2023) 'Solving the Challenges of HD Mapping for Smart Navigation in Autonomous Cars'. Intellias.

O'Hanlon, B.W. *et al.* (2010) 'Real-Time Spoofing Detection in a Narrow-Band Civil GPS Receiver', pp. 2211–2220. Available at: http://www.ion.org/publications/abstract.cfm?jp=p&articleID=9335 (Accessed: 15 June 2021).

O'Hanlon, B.W. *et al.* (2013) 'Real-Time GPS Spoofing Detection via Correlation of Encrypted Signals', *NAVIGATION, Journal of the Institute of Navigation*, 60(4), pp. 267–278. Available at: http://www.ion.org/publications/abstract.cfm?jp=j&articleID=102607 (Accessed: 15 June 2021).

Panice, G. *et al.* (2017) 'A SVM-based detection approach for GPS spoofing attacks to UAV', in *ICAC 2017 - 2017 23rd IEEE International Conference on Automation and Computing: Addressing Global Challenges through Automation and Computing*. Institute of Electrical and Electronics Engineers Inc. Available at: https://doi.org/10.23919/IConAC.2017.8081999.

PNT Program | Homeland Security (no date). Available at: https://www.dhs.gov/science-and-technology/pnt-program (Accessed: 26 September 2024).

Psiaki, M.L. *et al.* (2014) 'GNSS Spoofing Detection Using Two-Antenna Differential Carrier Phase', pp. 2776–2800. Available at: http://www.ion.org/publications/abstract.cfm?jp=p&articleID=12530 (Accessed: 15 June 2021).

Psiaki, M.L. and Humphreys, T.E. (2016) 'GNSS Spoofing and Detection', *Proceedings of the IEEE*, 104(6), pp. 1258–1270. Available at: https://doi.org/10.1109/JPROC.2016.2526658.

Qiao, J. *et al.* (2023) 'A survey of GNSS interference monitoring technologies', *Frontiers in Physics*, 11, p. 1133316. Available at: https://doi.org/10.3389/FPHY.2023.1133316/BIBTEX.

Ramanishka, V. *et al.* (2018) 'Toward Driving Scene Understanding: A Dataset for Learning Driver Behavior and Causal Reasoning', *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 7699–7707. Available at: http://arxiv.org/abs/1811.02307 (Accessed: 31 January 2021).

Ramesh, A.N. *et al.* (2021) 'Landmark-based RADAR SLAM for Autonomous Driving', *Proceedings International Radar Symposium*, 2021-June. Available at: https://doi.org/10.23919/IRS51887.2021.9466220.

Robusto, C.C. (1957) 'The Cosine-Haversine Formula', *The American Mathematical Monthly*, 64(1), p. 38. Available at: https://doi.org/10.2307/2309088.

Rohde, J. *et al.* (2016) 'Precise vehicle localization in dense urban environments', in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pp. 853–858.

Saastamoinen, J (1972) 'Atmospheric correction for the troposphere and stratosphere in radio ranging satellites', *The use of artificial satellites for geodesy*, 15, pp. 247–251.

Saastamoinen, J. (1972) 'Contributions to the theory of atmospheric refraction', *Bulletin Géodésique*, 46(3), pp. 279–298. Available at: https://doi.org/10.1007/BF02521844.

Saroufim, J., Hayek, S.W. and Kassas, Z.M. (2023) 'Simultaneous LEO satellite tracking and differential LEO-aided IMU navigation', in *2023 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, pp. 179–188.

Schaefer, A. *et al.* (2019) 'Long-term urban vehicle localization using pole landmarks extracted from 3-D lidar scans', in *2019 European Conference on Mobile Robots (ECMR)*, pp. 1–7.

Sefati, M. *et al.* (2017) 'Improving vehicle localization using semantic and pole-like landmarks', in *2017 IEEE Intelligent Vehicles Symposium (IV)*, pp. 13–19. Available at: https://doi.org/10.1109/IVS.2017.7995692.

Septentrio (2024) 'Spoofing Your GPS: How to Make Your GPS Attack-Proof'.

Shin, H.S. *et al.* (2014) 'Navigation-Aware Guidance in Unknown and GPS-Denied Environments Using the Information Theory'. Available at: https://doi.org/10.21535/proicius.2014.v10.258.

Some, E. and Gasiewski, A.J. (2023) 'Software Defined Radio Injection-Locking using a GPS signal for multichannel coherent receivers', *IEEE Aerospace Conference Proceedings*, 2023-March. Available at: https://doi.org/10.1109/AERO55745.2023.10115547.

Spilker, J.J. (1996) 'Interference effects and mitigation techniques', *Global Positioning System: Theory and applications.*, 1, pp. 717–771.

Spilker Jr, J.J. *et al.* (1996) *Global positioning system: theory and applications, volume I*. American Institute of Aeronautics and Astronautics.

'Springer Handbook of Global Navigation Satellite Systems' (2017) *Springer Handbook of Global Navigation Satellite Systems* [Preprint]. Available at: https://doi.org/10.1007/978-3-319-42928-1.

Sun, M. *et al.* (2017) 'GPS Spoofing Detection Based on Decision Fusion with a K-out-of-N Rule', *International Journal of Network Security*, 19(5), pp. 670–674. Available at: https://doi.org/10.6633/IJNS.201709.19(5).03.

Taketomi, T., Uchiyama, H. and Ikeda, S. (2017) 'Visual SLAM algorithms: A survey from 2010 to 2016', *IPSJ transactions on computer vision and applications*, 9, pp. 1–11.

Tanil, C. *et al.* (2016) 'Kalman filter-based INS monitor to detect GNSS spoofers capable of tracking aircraft position', in *Proceedings of the IEEE/ION Position, Location and Navigation Symposium, PLANS 2016*. Institute of Electrical and Electronics Engineers Inc., pp. 1027–1034. Available at: https://doi.org/10.1109/PLANS.2016.7479805.

Tanıl, Ç. *et al.* (2018) 'Experimental validation of INS monitor against GNSS spoofing', in *Proceedings of the 31st International Technical Meeting of the Satellite Division of the Institute of Navigation, ION GNSS+ 2018.* Institute of Navigation, pp. 2923–2937. Available at: https://doi.org/10.33012/2018.15902.

Tao, Q. *et al.* (2022) 'SeqPolar: Sequence matching of polarized LiDAR map with HMM for intelligent vehicle localization', *IEEE Transactions on Vehicular Technology*, 71(7), pp. 7071–7083.

Trimble (2024) 'Protection Against GNSS Spoofing'.

Two years since the Tesla GPS hack - GPS World : GPS World (no date). Available at: https://www.gpsworld.com/two-years-since-the-tesla-gps-hack/ (Accessed: 12 May 2022).

U.S. Department of Homeland Security (2021) 'GitHub - cisagov/PNT-Integrity: The PNT Integrity Library provides users a method to verify the integrity of the received GPS data and ranging signals, thereby improving resiliency against potential GPS signal loss.' Available at: https://github.com/cisagov/PNT-Integrity (Accessed: 28 July 2023).

Using Inertial Systems to Overcome GPS Spoofing - KVH Mobile World (no date). Available at: https://www.kvhmobileworld.kvh.com/using-inertial-systems-to-overcome-gps-spoofing/ (Accessed: 30 October 2020).

Vagle, N., Broumandan, A. and Lachapelle, G. (2017) 'Multi-antenna GNSS and INS/odometer coupling for robust vehicular navigation', in *Proc. Int. Tech. Symp. Navigat. Timing*, pp. 4816–4828.

Vorst, P. *et al.* (2008) 'Self-localization with RFID snapshots in densely tagged environments', in *2008 IEEE/RSJ international conference on intelligent robots and systems*, pp. 1353–1358.

Wang, C. *et al.* (2018) 'Vehicle localization at an intersection using a traffic light map', *IEEE transactions on intelligent transportation systems*, 20(4), pp. 1432–1441.

Warner, J.S. and Johnston, R.G. (2002) 'A simple demonstration that the global positioning system (GPS) is vulnerable to spoofing', *Journal of security administration*, 25(2), pp. 19–27.

Weng, L. *et al.* (2018) 'Pole-Based Real-Time Localization for Autonomous Driving in Congested Urban Scenarios', in *2018 IEEE International Conference on Real-time Computing and Robotics (RCAR)*, pp. 96–101. Available at: https://doi.org/10.1109/RCAR.2018.8621688.

Xin, S., Geng, J. and Hsu, L.T. (2024) 'Factor Graph Optimization-based GNSS PPP-RTK: An Alternative Platform to Study Urban GNSS Precise Positioning', *IEEE Transactions on Aerospace and Electronic Systems* [Preprint]. Available at: https://doi.org/10.1109/TAES.2024.3360380.

Yang, J. *et al.* (2013) 'Detection and localization of multiple spoofing attackers in wireless networks', *IEEE Transactions on Parallel and Distributed Systems*, 24(1), pp. 44–58. Available at: https://doi.org/10.1109/TPDS.2012.104.

Yang, X. *et al.* (2021) 'A novel NLOS error compensation method based IMU for UWB indoor positioning system', *IEEE Sensors Journal*, 21(9), pp. 11203–11212.

Yousif, K., Bab-Hadiashar, A. and Hoseinnezhad, R. (2015) 'An overview to visual odometry and visual SLAM: Applications to mobile robotics', *Intelligent Industrial Systems*, 1(4), pp. 289–311.

Yu, Z. *et al.* (2024) 'A lightweight odometry network for GNSS/INS integration during GNSS outages', *Applied Soft Computing*, 151, p. 111143. Available at: https://doi.org/https://doi.org/10.1016/j.asoc.2023.111143.

Zeng, K. *et al.* (2017) 'A practical GPS location spoofing attack in road navigation scenario', in *HotMobile 2017 - Proceedings of the 18th International Workshop on Mobile Computing Systems and Applications*. New York, NY, USA: Association for Computing Machinery, Inc, pp. 85–90. Available at: https://doi.org/10.1145/3032970.3032983.

Zeng, K. *et al.* (2018) *All Your GPS Are Belong To Us: Towards Stealthy Manipulation of Road Navigation Systems*. Available at: https://www.usenix.org/conference/usenixsecurity18/presentation/zeng (Accessed: 19 June 2021).

Zhang, K. and Papadimitratos, P. (2019) 'On the Effects of Distance-decreasing Attacks on Cryptographically Protected GNSS Signals', *ION 2019 International Technical Meeting Proceedings*, pp. 363–372. Available at: https://doi.org/10.33012/2019.16736.

Zhang, L. *et al.* (2022) 'Enhanced GNSS spoofing detector via multiple-epoch inertial navigation sensor prediction in a tightly-coupled system', *IEEE Sensors Journal*, 22(9), pp. 8633–8647.

Zhao, C. *et al.* (2023) 'Analysis of baseline impact on differential doppler positioning and performance improvement method for LEO opportunistic navigation', *IEEE Transactions on Instrumentation and Measurement*, 72, pp. 1–10.

Zidan, J., Adegoke, Elijah I, *et al.* (2020) 'GNSS vulnerabilities and existing solutions: A review of the literature', *IEEE Access*, 9, pp. 153960–153976.

Zidan, J., Adegoke, E. I., *et al.* (2020) 'GNSS Vulnerabilities and Existing Solutions: A Review of the Literature', *IEEE Access*, pp. 1–1. Available at: https://doi.org/10.1109/access.2020.2973759.

Ziparo, V.A. *et al.* (2007) 'RFID-based exploration for large robot teams', in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, pp. 4606–4613.