Technical Report Documentation Page

1. Report No.	2. Government Accession No. N/A	3. Recipient's Catalog No. N/A
4. Title and Subtitle Data-Driven Modeling and Anomaly Detection of Attacked HATS		5. Report Date September 17, 2026 6. Performing Organization Code N/A
7. Author(s) Abdollah Homaifar, PhD https://orcid.org/0000-0003-1179-3221		8. Performing Organization Report No. N/A
9. Performing Organization Name and Address North Carolina Agricultural and Technical State University, 1601 E Market St, Greensboro, NC 27411		10. Work Unit No. (TRAIS) N/A 11. Contract or Grant No. 69A3552348327
12. Sponsoring Agency Name and Address CARMEN+ University Transportation Center The Ohio State University, 930 Kinnear Rd., Columbus, OH 43212		13. Type of Report and Period Covered Final (Nov. 1, 2023 to August 30, 2026) 14. Sponsoring Agency Code N/A
15. Supplementary Notes Conducted in cooperation with the U.S. Department of Transportation, Federal Highway Administration.		

16. Abstract

This project developed data-driven methods to improve the cybersecurity, verification and validation (V&V), and resilience of Highly Automated Transportation Systems (HATS) under cyberattack. Subtask 1 focused on Global Navigation Satellite System (GNSS) spoofing detection using onboard vehicle measurements, adaptive DBSCAN, and time-series analysis, and extended the work toward stereo visual navigation when GNSS becomes untrustworthy. The data-driven detector reduced the nominal displacement threshold from 3.35 m to 1.79 m; adaptive DBSCAN detected below-threshold multiple small biased attacks with 98.38% accuracy; and the time-series framework detected 0.28 m spoofing steps with 93.66% accuracy. The ongoing visual-navigation framework achieved 88.50% and 88.00% direction-classification accuracy for the two global displacement axes and a 7.12 mm translational relative-pose root-mean-square error. Subtask 2 developed a fuzzy-equivalence V&V framework that produced a 67.24% Adjusted Accuracy and 89.57% estimated peak accuracy benchmark, and a GNSS onset-localization framework whose enhanced neural model achieved zero-sample median error and 4.75-sample mean absolute error at a sampling rate of 10 Hz.

17. Key Words		18. Distribution Statement	
Highly Automated Transportation Systems, Transportation Cybersecurity, GNSS Spoofing, Anomaly Detection, Machine Learning, Verification and Validation, Visual Navigation, Resilient Navigation		No restrictions. This document is available to the public through NTIS: National Technical Information Service Springfield, Virginia 22161	
19. Security Classif.(of this report)	**20. Security Classif.(of this page)**	**21. No. of Pages**	**22. Price**
Unclassified	Unclassified	117	N/A

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

CARMEN+

Center for Automated Vehicles Research with Multimodal Assured Navigation

USDOT University Transportation Centers Program

Final Report: Data-Driven Modeling and Anomaly Detection of Attacked Highly Automated Transportation Systems (HATS)

P.I.	Project Info:
Abdollah Homaifar North Carolina Agriculture and Technical State University Electrical and Computer Engineering	Grant No. 69A3552348327
	DUNS: 832127323
	EIN #: 31-6025986
	Project Effective: Nov. 1, 2023 Project End: August 30, 2026 Submission: Sep. 17, 2026

Consortium Members:

The Ohio State University College of Engineering | TEXAS The University of Texas at Austin | UCI University of California, Irvine | North Carolina Agricultural and Technical State University

DISCLAIMER

The contents of this report reflect the views of the authors, who are responsible for the facts and accuracy of the information presented herein. This document is disseminated in the interest of information exchange. The report is funded, partially or entirely, under grant number 69A3552348327 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.

Table of Contents

Abstract 7
Acknowledgements 8
Executive Summary 9
1. Introduction and Problem Statement 11
1.1 Highly Automated Transportation Systems and Cybersecurity 11
1.2 Cyberattacks Against HATS 11
1.3 Need for Data-Driven Cybersecurity, Verification, and Validation 12
1.4 Project Research Framework 13
1.5 CARMEN+ and USDOT Relevance 15
1.6 Organization of the Report 15
2. Project Objectives 16
2.1 Overall Project Objective 16
2.2 Subtask 1: Detection and Identification 16
2.2.1 Detect Conventional and Stealthy GNSS Spoofing Attacks 16
2.2.2 Develop Data-Driven Displacement-Based Attack Detection 16
2.2.3 Develop Adaptive Anomaly-Detection Mechanisms 16
2.2.4 Detect Multiple Small Biased Attacks Using Temporal Information 16
2.2.5 Investigate Navigation Capability Following GNSS Compromise or Denial 17
2.3 Subtask 2: Data-Driven Verification and Validation of Attacked HATS 17
2.3.1 Data-Driven Validation of Anomaly Detection Performance 17
2.3.2 GNSS Spoofing Onset Localization in Nonlinear State Estimators 19
2.4 Relationship Between the Two Subtasks 21
3. Technical Approach and Methodology 21
3.1 Subtask 1: Detection and Identification 21
3.1.1 Contribution 1: Data-Driven GNSS Spoofing Detection 21
3.1.2 Contribution 2: Adaptive GNSS Spoofing Detection Using DBSCAN 33
3.1.3 Contribution 3: Time-Series-Based Detection of Multiple Small Biased GNSS Spoofing Attacks 39
3.1.4 Contribution 4: Stereo Visual Navigation in GNSS-Compromised and Denied Environments 48
3.2 Subtask 2: Data-Driven Verification and Validation of Attacked HATS 57

3.2.1 Contribution 1: Fuzzy Equivalence Relation Framework for Anomaly Detection Accuracy Estimation 58
3.2.2 Contribution 2: Onset Localization Under Deceptive GNSS in Nonlinear State Estimators 62
4. Results and Findings 69
4.1 Subtask 1: Detection and Identification 69
4.1.1 Contribution 1: ICAIC Data-Driven Detection Results 69
4.1.2 Contribution 2: Adaptive DBSCAN Results 74
4.1.3 Contribution 3: Time-Series Detection Results 77
4.1.4 Contribution 4: Stereo Visual Navigation Results 81
4.2 Subtask 2: Data-Driven Verification and Validation of Attacked HATS 89
4.2.1 Contribution 1: Fuzzy Equivalence Relation Validation Results 89
4.2.2 Contribution 2: GNSS Spoofing Onset Localization Results 93
5. Implications for Transportation 97
5.1 Overall Transportation Significance 97
5.2 Subtask 1: Detection and Identification 98
5.2.1 Improved Cyberattack Detection for Automated Transportation 98
5.2.2 Practical Use of Existing Onboard Vehicle Sensors 99
5.2.3 Resilient Localization and Navigation Under GNSS Compromise 100
5.2.4 Transportation Safety and Deployment Implications 100
5.3 Subtask 2: Data-Driven Verification and Validation of Attacked HATS 101
5.3.1 Improving Confidence in Data-Driven Cybersecurity Models 101
5.3.2 Verification of Cyberattack Timing and State-Estimation Integrity 102
5.3.3 Implications for Recovery and Resilient Navigation 102
5.3.4 Broader V&V Implications for Transportation Cybersecurity 103
5.4 Combined Impact on HATS Safety and Resilience 104
6. Workforce Development 105
6.1 Graduate Student Involvement 105
6.3 Research Training and Technical Skills 105
6.4 Student Publications and Conference Participation 105
6.5 Contribution to Transportation Cybersecurity Workforce Development 106
7. Publications and Presentations 106
8. Data Management and Dataset Statement 108
8.1 Public Datasets Used in the Project 108

8.2 Project-Generated and Derived Research Data 109
8.3 Visual Navigation Experimental Dataset 109
8.4 Data Storage, Access, and Archiving 109
9. Matching Funds Statement 109
10. Conclusions and Recommendations 111
10.1 Conclusions 111
10.2 Recommendations and Future Research 112
References 114

Abstract

This project developed data-driven methods to improve the cybersecurity, verification and validation (V&V), and resilience of Highly Automated Transportation Systems (HATS) under cyberattack. Subtask 1 focused on Global Navigation Satellite System (GNSS) spoofing detection using onboard vehicle measurements, adaptive DBSCAN, and time-series analysis, and extended the work toward stereo visual navigation when GNSS becomes untrustworthy. The data-driven detector reduced the nominal displacement threshold from 3.35 m to 1.79 m; adaptive DBSCAN detected below-threshold multiple small biased attacks with 98.38% accuracy; and the time-series framework detected 0.28 m spoofing steps with 93.66% accuracy. The ongoing visual-navigation framework achieved 88.50% and 88.00% direction-classification accuracy for the two global displacement axes and a 7.12 mm translational relative-pose root-mean-square error. Subtask 2 developed a fuzzy-equivalence V&V framework that produced a 67.24% Adjusted Accuracy and 89.57% estimated peak accuracy benchmark, and a GNSS onset-localization framework whose enhanced neural model achieved zero-sample median error and 4.75-sample mean absolute error at a sampling rate of 10 Hz.

Acknowledgements

This research was supported by the CARMEN+ University Transportation Center (UTC), funded by the U.S. Department of Transportation University Transportation Centers Program under Grant 69A3552348327. This work was also partially supported by NASA-ULI under Cooperative Agreement Nos. 80NSSC20M0161 and 80NSSC25M7098, the National Science Foundation (NSF) under Grant 2301553, and the North Carolina Department of Transportation (NCDOT) AAM and UAS Program under Grant RP2025-43.

Executive Summary

Highly Automated Transportation Systems (HATS) depend on accurate sensing, localization, state estimation, and automated decision making. As vehicles become more connected and automated, cyberattacks can affect physical transportation safety as well as information systems. Global Navigation Satellite System (GNSS) spoofing is a particular concern because false but plausible navigation information can produce an incorrect vehicle position or motion state. Sophisticated attacks may also evolve gradually, allowing individual manipulations to remain close to normal sensor uncertainty. This project addressed these risks through two coordinated subtasks focused on detecting and identifying compromised navigation information and verifying the reliability of the data-driven methods and system information used under attack.

Subtask 1 progressively strengthened the ability to detect GNSS spoofing as attacks became more difficult to distinguish from normal navigation uncertainty. The initial approach predicted short-horizon displacement from vehicle speed and gyroscope measurements independently of GNSS and compared that prediction with GNSS-reported motion. It reduced the nominal displacement-error threshold from 3.35 m for the analytical baseline to 1.79 m, while detecting multiple small biased attacks with 97.83% accuracy and 100% recall. This tighter uncertainty boundary improved sensitivity to smaller malicious trajectory deviations, but individual attacks could still remain below the 1.79 m threshold.

To detect individual attack manipulations below the 1.79 m threshold, an adaptive **D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise (DBSCAN) method replaced the fixed decision boundary with one updated from the recursive statistics of residuals accepted as normal. It achieved 98.38% accuracy and 100% sensitivity for attacks with approximately 1.41 m steps. As the attack magnitude decreased further, individual manipulations became increasingly difficult to distinguish from nominal displacement uncertainty. The research therefore incorporated temporal evidence, using the evolution of displacement errors over time to identify attack behavior that may not be distinguishable from nominal behavior at an individual time step. The resulting time-series framework achieved 95.92% accuracy for approximately 0.56 m attack steps over 25 s and 93.66% accuracy for approximately 0.28 m attack steps over 50 s. The 0.28 m manipulation corresponds to approximately 1.5 standard deviations of nominal displacement error, demonstrating detection of individual spoofing increments close to normal system uncertainty. Together, these methods establish a progression from direct consistency checking to adaptive statistical detection and temporal inference as attacks become increasingly stealthy.

Detecting that GNSS has become untrustworthy, however, does not provide a replacement navigation source. The project therefore extended Subtask 1 toward GNSS-independent motion estimation using stereo vision. The ongoing framework estimates short-horizon planar vehicle displacement from stereo imagery and achieved direction-classification accuracies of 88.50% and 88.00% for the two global displacement axes. At the trajectory level, it achieved a translational Relative Pose Error root-mean-square error of 7.12 mm, while the Absolute Trajectory Error reached 210.92 mm because short-horizon errors accumulate during trajectory integration. These results demonstrate the feasibility of independent visual motion estimation while also identifying the remaining need for full-scale outdoor validation, execution-time characterization, visual-inertial integration, and improved robustness to dynamic environments.

Subtask 2 addressed a complementary V&V question arising from the increasing use of data-driven methods and the difficulty of distinguishing subtle attacks from nominal behavior: whether the data, models, and inferred system history used in cybersecurity decisions are sufficiently reliable. The first investigation examined whether the structure and separability of labeled cyberattack data could indicate the level of detection performance that a learning-based model could reasonably achieve. A fuzzy-equivalence framework was developed to estimate expected anomaly-detection performance from dataset structure rather than relying only on the observed accuracy of a trained classifier. Applied to CIC-DDoS2019, the framework produced a 67.24% Adjusted Accuracy and an estimated peak accuracy benchmark of 89.57%. Although dataset-specific, these measures provide a means of assessing whether the available data support sufficient separation between benign and malicious behavior before a data-driven detector is treated as reliable.

The second question concerns the delay between attack onset and detection, during which malicious measurements may already have affected the navigation estimator. The onset-localization investigation therefore addressed the gap between the true attack onset and alarm time by formulating onset localization as a retrospective change-point problem in nonlinear-estimator residuals. Across 18,034 held-out spoofing realizations, an enhanced neural method achieved a zero-sample median onset error, a 4.75-sample mean absolute error, and a 9.58-sample root-mean-square error. Performance degraded for low-energy and gradually evolving ramp attacks, identifying conditions under which onset localization is more difficult. The estimated onset provides information about which recent navigation states may already have been compromised before detection, establishing a foundation for subsequent state reconstruction.

The completed research improved detection of increasingly subtle GNSS spoofing, provided methods for evaluating the reliability of data-driven cybersecurity decisions and determining when estimator corruption likely began, and demonstrated the feasibility of GNSS-independent visual motion estimation. These results support CARMEN+ and U.S. Department of Transportation efforts to reduce transportation cybersecurity risks and improve the resilience of automated transportation systems. Further development should focus on broader vehicle-based validation, integration of attack-onset estimates with navigation-state recovery, and maturation of visual-inertial navigation for operation when GNSS is compromised or unavailable.

1. Introduction and Problem Statement

1.1 Highly Automated Transportation Systems and Cybersecurity

Highly Automated Vehicles (HAVs) are increasingly integrated into modern transportation systems because they can support improved mobility, operational efficiency, driving consistency, and transportation safety [1], [2]. Their operation depends on coordinated perception, localization, state estimation, decision making, planning, and control. As these capabilities become more automated and interconnected, HAVs become important components of Highly Automated Transportation Systems (HATS), in which physical vehicle motion is governed by software, sensing, communications, and computational decision processes. Safe operation therefore depends not only on mechanical reliability, but also on the integrity and availability of the information used by the automated system.

HAVs rely on heterogeneous sensing architectures to estimate both the surrounding environment and the vehicle state. Common sensing modalities include cameras, Light Detection and Ranging (LiDAR), Radio Detection and Ranging (RADAR), Inertial Measurement Units (IMUs), wheel-speed sensors, and Global Navigation Satellite System (GNSS) receivers [2], [3]. These sensors provide complementary information for obstacle detection, environmental perception, motion estimation, localization, and navigation. GNSS is particularly important because it provides globally referenced positioning information, while onboard inertial, wheel-speed, and visual sensors provide independent measurements of physical vehicle motion. The reliability of automated transportation consequently depends on maintaining consistency among multiple information sources and identifying when one of those sources can no longer be trusted.

The same sensing, communication, and computational integration that enables higher levels of automation also expands the cybersecurity attack surface. Malicious modification of a sensor measurement, communication stream, or estimator input can alter the vehicle state perceived by the automated system and subsequently influence planning and control decisions [4], [5]. In HATS, cybersecurity is therefore a transportation-safety concern rather than only an information-security concern. A successful attack can propagate from manipulated data to physical vehicle behavior, making cyberattack detection, model assurance, navigation integrity, and resilient operation essential elements of trustworthy HAV deployment.

This project addresses these requirements through a data-driven cybersecurity and assurance framework. The research considers how attacks can be detected from inconsistencies among vehicle information sources, how the reliability of data-driven anomaly-detection models can be assessed, how the true onset of an attack can be reconstructed after an alarm, and how vehicle motion may continue to be estimated when GNSS becomes unavailable or untrustworthy. These complementary questions reflect the broader need for HATS to recognize compromised information, evaluate the confidence associated with cybersecurity decisions, and preserve transportation safety under attacked conditions.

1.2 Cyberattacks Against HATS

Cyberattacks against HATS can target sensing, communication, estimation, and decision-making functions. An attacker may attempt to reduce system availability, inject misleading information,

manipulate measurements, or exploit the dependence of automated functions on externally supplied data [4]-[7]. Distributed Denial of Service (DDoS) attacks are one example of attacks that can disrupt cyber infrastructure and communications, while sensor-oriented attacks can directly corrupt the measurements used to estimate vehicle motion or environmental state. These threats are especially consequential in automated transportation because a cyber-domain disturbance can influence a safety-critical physical process.

GNSS spoofing represents a particularly important sensor-integrity threat. In a spoofing attack, counterfeit or manipulated satellite-navigation information causes the receiver to report an incorrect position or motion state while the signal may continue to appear plausible to the navigation system [10]-[12]. Conventional attacks can produce substantial discrepancies, such as false turn-by-turn motion, false movement of a stopped vehicle, or false stationary behavior of a moving vehicle. More sophisticated attacks can be mounted gradually through a sequence of small biased manipulations. Each individual manipulation may remain within nominal sensor uncertainty or below a conventional threshold even though the repeated sequence produces a large accumulated localization error.

Stealthiness can also arise from the temporal behavior of the monitoring process itself. Sequential integrity monitors generally require evidence to accumulate before an alarm is generated. As a result, the alarm time may occur after the true attack onset, and corrupted GNSS measurements may already have propagated through a recursive navigation estimator before the attack is formally declared [26]. Gradually evolving ramp-style attacks are especially challenging because their early residual signatures can be weak. This creates a vulnerable interval in which the system may still be using state estimates influenced by malicious measurements even though the integrity monitor has not yet accumulated sufficient evidence to trigger.

The consequences of undetected or poorly characterized attacks can propagate through localization, situational awareness, route selection, trajectory planning, and control. Corrupted positioning information can cause the automated system to reason from an incorrect vehicle state, while unreliable cybersecurity models can create false confidence in whether an observed condition is benign or malicious. Protecting HATS therefore requires more than a binary attack alarm. The system must detect both evident and stealthy anomalies, understand the reliability of the models used to make cybersecurity decisions, determine when corruption may have begun, and support a transition to trusted sensing when compromised information should no longer be used.

1.3 Need for Data-Driven Cybersecurity, Verification, and Validation

The complexity of HATS makes comprehensive cybersecurity assessment difficult using only fixed analytical models or rule-based thresholds. Vehicle motion is nonlinear and time varying, sensor measurements contain noise and bias, and nominal residual distributions can change with operating conditions. Cyberattacks can exploit these characteristics by producing behavior that is statistically close to normal operation at individual time steps. Data-driven methods are valuable in this setting because they can learn relationships directly from operational data, characterize multivariable uncertainty, and identify temporal or statistical patterns that are difficult to represent with a single fixed model [4], [5].

The first research direction required by this project is detection and identification. Under Subtask 1, trusted onboard measurements are used to evaluate GNSS consistency through progressively more capable methods. Data-driven displacement prediction reduces dependence on accumulated analytical position reconstruction [10]; adaptive DBSCAN allows the anomaly boundary to evolve with nominal residual statistics [11]; and time-series analysis uses recent residual history to detect multiple small biased attacks whose individual steps can remain below conventional thresholds [12]. The ongoing stereo visual navigation work extends this direction beyond detection by investigating short-horizon planar displacement prediction from onboard visual information when GNSS can no longer be treated as a trusted localization source [23].

The second research direction is data-driven verification and validation (V&V). High reported accuracy from an anomaly detector does not by itself establish that the underlying data provide reliable separation between benign and malicious behavior. The fuzzy equivalence relation contribution addresses this issue by analyzing intra-class cohesion and inter-class ambiguity and by using an alpha-cut-based Adjusted Accuracy measure to establish a dataset-specific performance benchmark for data-driven anomaly-detection models [25]. This provides a V&V perspective before or alongside model deployment by asking whether the structure of the available data supports the performance expected from the detector.

V&V is also required after an attack alarm. The GNSS onset-localization contribution treats the interval between true attack initiation and eventual detection as a retrospective change-point problem in nonlinear estimator residuals [26]. By accounting for temporal correlation, attack-induced residual energy, and different attack dynamics, the framework evaluates whether the true onset can be reconstructed and how much uncertainty remains in that estimate. This capability complements attack detection because it identifies the portion of recent estimator history that may already have been affected before the alarm.

Together, these two research directions support a broader assurance sequence for attacked HATS. Detection and identification determine whether localization information has become inconsistent or malicious; V&V evaluates whether the cybersecurity evidence and associated models are sufficiently trustworthy; onset localization identifies when estimator corruption likely began; and alternative sensing provides a pathway toward continued navigation after GNSS is rejected. This integrated perspective is more appropriate for safety-critical transportation than treating attack detection as an isolated classification task.

1.4 Project Research Framework

The research conducted under this project is organized into two complementary subtasks that address different stages of cybersecurity assurance and resilience in HATS. Subtask 1 focuses on Detection and Identification, with emphasis on GNSS spoofing and the transition toward post-detection visual navigation. Subtask 2 focuses on Data-Driven Verification and Validation of Attacked HATS, with one contribution addressing the expected reliability of anomaly-detection models and a second contribution addressing retrospective localization of deceptive GNSS attack onset. The two subtasks are related by a common objective: improving the trustworthiness of the information used by automated transportation systems before, during, and after a cyberattack.

1.4.1 Subtask 1: Detection and Identification

Subtask 1 develops a progression of methods for identifying compromised GNSS information from measurements that remain independent of the potentially attacked navigation signal. The first contribution uses vehicle speed and gyroscope information in a data-driven displacement predictor and compares the resulting motion estimate with GNSS-derived displacement and speed [10]. The second contribution extends this framework with adaptive DBSCAN so that the anomaly boundary is updated from recursively estimated statistics of residuals accepted as nominal [11]. The third contribution incorporates explicit temporal memory through time-series analysis and an attack-probability predictor to detect smaller and more gradually accumulating spoofing biases [12].

The fourth contribution extends the research from attack recognition toward navigation after GNSS compromise or denial. The ongoing stereo visual navigation framework uses sparse stereo correspondence, chained temporal tracking, a compact 101-dimensional geometric feature representation, direction classification, and conditional Extreme Gradient Boosting (XGBoost) prediction to estimate short-horizon planar global displacement from stereo images [23]. This work is a research prototype and is not presented as a completed full-scale replacement for GNSS. Its role in the project is to establish a pathway from identifying an untrustworthy localization source toward maintaining motion information using independent onboard sensing.

1.4.2 Subtask 2: Data-Driven Verification and Validation of Attacked HATS

Subtask 2 develops two V&V contributions that complement the attack-detection work. The first contribution uses fuzzy tolerance and fuzzy equivalence relations to evaluate the structure of labeled cyber-traffic data and estimate the performance that a data-driven anomaly detector can reasonably be expected to achieve [25]. Rather than proposing another DDoS classifier, the method evaluates intra-class similarity, inter-class ambiguity, false-alarm behavior, and an Adjusted Accuracy metric across different alpha-cuts. The resulting benchmark provides an independent reference for interpreting detector performance and illustrates how dataset structure can be incorporated into data-driven model validation.

The second contribution addresses GNSS spoofing after an integrity monitor has raised an alarm. Because recursive navigation estimators may already have incorporated corrupted measurements before detection, the work formulates spoofing-onset localization as a retrospective change-point estimation problem in the residual sequence of a nonlinear state estimator [26]. A correlation-aware Generalized Likelihood Ratio localizer and neural temporal localizers are evaluated against a Cumulative Sum baseline, while the effects of residual spoof energy and step-versus-ramp attack dynamics are examined. This contribution extends V&V from evaluating model performance to assessing the timing and integrity implications of a detected navigation attack.

Both contributions emphasize that cybersecurity decisions in HATS should be accompanied by quantitative evidence of confidence. The fuzzy framework asks whether the data support reliable anomaly discrimination before a detector is trusted, while the onset framework asks how reliably the beginning of a detected GNSS attack can be reconstructed from estimator evidence. These capabilities provide information that can support future model acceptance, integrity monitoring, estimator recovery, and resilient navigation decisions.

1.5 CARMEN+ and USDOT Relevance

The research conducted in this project directly supports the CARMEN+ University Transportation Center emphasis on reducing transportation cybersecurity risks and improving the safety and resilience of automated transportation systems [13]. The CARMEN+ research framework addresses cybersecurity threats to connected and automated transportation, Positioning, Navigation, and Timing (PNT) reliability, cyber-resilient mitigation, and testing and validation under jammed, spoofed, or otherwise compromised conditions. The project contributes to these priorities by combining cyberattack detection, model V&V, post-alarm attack-onset analysis, and research toward navigation without trusted GNSS.

Subtask 1 supports these objectives by improving the ability of HATS to recognize compromised localization information using existing onboard sensing and by addressing both conventional and stealthy GNSS spoofing [10]-[12]. The ongoing visual-navigation contribution further investigates how vehicle motion can be estimated after GNSS is rejected [23]. Subtask 2 complements this work by evaluating the expected reliability of data-driven anomaly-detection models and by reconstructing the time at which deceptive GNSS measurements began to influence a nonlinear state estimator [25], [26]. These activities address not only whether an attack can be detected, but also whether the associated cybersecurity evidence can be trusted and how the affected estimator history can be interpreted.

Subtask 1 contributes GNSS spoofing detection and stereo-based motion estimation, while Subtask 2 contributes dataset-based model validation and attack-onset localization. These completed studies address the CARMEN+ and USDOT objectives of transportation cybersecurity assessment and resilient automated mobility. Vehicle-scale validation, edge implementation, estimator recovery, and integration of the methods remain future work.

The completed methods may reduce the risk that corrupted navigation information is accepted without adequate detection or evaluation before it influences automated-vehicle decisions.

1.6 Organization of the Report

The remainder of this report follows the two-subtask project structure. Chapter 2 defines the overall project objectives and the specific objectives for Subtask 1, Detection and Identification, and Subtask 2, Data-Driven Verification and Validation of Attacked HATS. Chapter 3 presents the technical approaches, including the three GNSS spoofing-detection contributions, the ongoing stereo visual navigation framework, the fuzzy equivalence relation validation framework, and the GNSS spoofing onset-localization framework. Chapter 4 presents and analyzes the corresponding experimental results and explicitly distinguishes demonstrated performance from ongoing or future capabilities. Chapter 5 discusses the transportation implications of each subtask and synthesizes their combined contribution to HATS safety and resilience. The subsequent chapters document workforce development, publications and presentations, data management, matching funds, and the overall conclusions and recommendations.

2. Project Objectives

2.1 Overall Project Objective

The overall objective of the project is to develop data-driven methods that improve the cybersecurity, safety assessment, verification, and validation of HATS exposed to cyberattacks. The project addresses the need to detect and identify malicious manipulation of vehicle information while also supporting the assessment of system behavior and reliability under attacked conditions. These objectives are consistent with the original CARMEN+ project scope, which emphasizes detectability, identifiability, and data-driven evaluation of attacked vehicle states.

2.2 Subtask 1: Detection and Identification

Subtask 1 focuses on developing practical methods for detecting and identifying cyberattacks affecting vehicle localization and navigation, with particular emphasis on GNSS spoofing. Its objectives span conventional and stealthy spoofing detection, data-driven displacement estimation, adaptive anomaly detection, temporal analysis of persistent low-magnitude attacks, and continued navigation when GNSS becomes untrustworthy. Collectively, these objectives address increasingly challenging attack conditions while maintaining consideration of real-time implementation and vehicle safety.

2.2.1 Detect Conventional and Stealthy GNSS Spoofing Attacks

The detection scope encompasses both conventional GNSS spoofing attacks and stealthy attacks designed to evade conventional detection mechanisms. Particular attention is given to gradual and multiple small biased attacks in which individual manipulations may remain within nominal sensor uncertainty or below fixed detection thresholds while accumulating into significant localization errors over time.

2.2.2 Develop Data-Driven Displacement-Based Attack Detection

Data-driven models are pursued to estimate short-term vehicle displacement using trusted in-vehicle measurements and compare the estimated motion with GNSS-reported behavior. By capturing nonlinear relationships among vehicle motion variables, these models are intended to improve sensitivity to spoofing-induced inconsistencies compared with purely analytical motion models.

2.2.3 Develop Adaptive Anomaly-Detection Mechanisms

Adaptive anomaly-detection methods are developed to reduce dependence on fixed detection thresholds by allowing decision boundaries to respond to the statistical characteristics of incoming sensor data. This adaptability is intended to accommodate variations in sensor uncertainty, vehicle dynamics, and operating conditions that can reduce the reliability of static detection criteria.

2.2.4 Detect Multiple Small Biased Attacks Using Temporal Information

Temporal information is incorporated into the detection process to identify small but persistent anomalies whose individual deviations may not be sufficient to trigger conventional detection criteria. Rather than relying only on instantaneous deviations, time-series-based analysis

accumulates evidence of gradually evolving spoofing behavior to improve the detection of attacks designed to remain below conventional thresholds.

2.2.5 Investigate Navigation Capability Following GNSS Compromise or Denial

The scope extends beyond attack detection to investigate how navigation capability can be maintained when GNSS measurements become compromised, degraded, denied, or otherwise untrustworthy. Trusted onboard sensing, particularly visual information, is investigated for estimating short-horizon vehicle motion and supporting continued navigation without reliance on GNSS. Consistent with this objective, the Year 3 project scope expanded the experimental effort toward visual sensing and localization under GPS-compromised conditions.

2.3 Subtask 2: Data-Driven Verification and Validation of Attacked HATS

Subtask 2 focuses on data-driven verification and validation (V&V) of cybersecurity mechanisms and navigation-state information used in HATS under cyberattack. The research addresses reliability at two complementary stages. Before a data-driven anomaly-detection model is relied upon, the available data must provide sufficient distinction between benign and malicious behavior to support the expected level of detection performance. Following detection of a deceptive Global Navigation Satellite System (GNSS) attack, the integrity of the affected navigation history must also be assessed to determine when the attack began and identify the interval over which recursive state estimates may have been corrupted [25], [26].

These V&V needs are addressed through two complementary research directions. Data-driven anomaly-detection performance is evaluated using fuzzy equivalence relations to characterize the structure and separability of labeled cyberattack data and establish an independent performance benchmark [25]. GNSS spoofing onset localization addresses post-alarm navigation integrity by retrospectively estimating the attack start time from the residual sequence of a nonlinear state estimator [26]. Together, these directions extend V&V from assessing the expected reliability of data-driven detection to evaluating the temporal integrity of navigation-state information following cyberattack.

2.3.1 Data-Driven Validation of Anomaly Detection Performance

The anomaly-detection validation research examines whether the structure of labeled cyberattack data can provide an independent basis for assessing the performance that a data-driven detector can reasonably be expected to achieve. Presented at the 2025 IFSA Winter Conference on Automation, Robotics & Communications for Industry 4.0/5.0 (ARCI 2025) [25], the proposed fuzzy-equivalence framework differs from a conventional DDoS detector because its purpose is not to train another classifier and report its test accuracy. Instead, it analyzes similarity relationships among benign and DDoS samples to characterize class distinguishability and establish a performance reference against which subsequently trained detection models can be evaluated. The objectives therefore address achievable anomaly-detection performance, inter-class and intra-class distinguishability, validation metrics, and the establishment of a dataset-specific performance benchmark.

2.3.1.1 Estimate Achievable Anomaly-Detection Performance

The framework seeks to estimate the achievable performance of a data-driven anomaly-detection model before a trained model's reported accuracy is treated as sufficient evidence of reliability. It evaluates the intrinsic similarity structure of a labeled DDoS dataset and uses that structure to infer the level of class separation that an anomaly-detection model can reasonably be expected to exploit [25].

This analysis addresses a practical V&V gap identified in the study: many data-driven DDoS detection methods report high accuracy after training, but the dataset itself is rarely analyzed to establish an independent performance reference. The proposed approach therefore provides a baseline against which the observed performance of a subsequently trained detector can be judged, rather than assuming that a single reported accuracy value is sufficient to establish model reliability [25].

2.3.1.2 Quantify Inter-Class and Intra-Class Distinguishability

The framework also examines how clearly benign and DDoS samples can be distinguished within the available cyber-traffic data. Fuzzy tolerance and fuzzy equivalence relations are constructed to represent pairwise similarity among samples, allowing both intra-class similarity and inter-class overlap to be evaluated [25].

High intra-class similarity indicates that samples belonging to the same class exhibit coherent structure, whereas inter-class similarity indicates ambiguity between benign and malicious traffic. Varying the fuzzy α-cut changes the minimum similarity required for samples to remain associated, thereby revealing the tradeoff between within-class cohesion and between-class separation. The analysis seeks a region in which the two classes are sufficiently distinct to support reliable anomaly detection without discarding excessive within-class structure [25].

2.3.1.3 Develop an Adjusted Accuracy Metric for Model Validation

To incorporate class ambiguity into model validation, the study develops an Adjusted Accuracy metric that reflects both correctly grouped samples and overlap between the two traffic classes. The metric combines the dominant DDoS and benign cluster sizes while penalizing samples that remain mixed between the two classes, referred to in the study as tossed-up samples [25].

Adjusted Accuracy is intended to capture information that conventional classification accuracy alone does not represent. High nominal accuracy can coexist with substantial class overlap or an elevated false-alarm tendency near the decision boundary. By explicitly incorporating the inter-class ambiguity identified through fuzzy clustering, the metric provides a more conservative reference for evaluating whether the structure of a dataset supports reliable separation of benign and DDoS traffic [25].

2.3.1.4 Establish a Performance Benchmark for Data-Driven DDoS Detection Models

The fuzzy similarity analysis is ultimately used to establish a practical benchmark for evaluating data-driven DDoS detection models trained on the same dataset. The study varies the α-cut and examines the resulting cluster composition, false-alarm behavior, conventional performance measures, and Adjusted Accuracy to identify a balanced operating point rather than selecting a threshold solely because it produces the largest apparent accuracy [25].

For the CIC-DDoS2019 data examined in the study, an α-cut of 95.5% is identified as the operating point associated with an Adjusted Accuracy of 67.24% and an expected peak accuracy of 89.57% [25]. These values are not interpreted as universal DDoS-detection limits. Rather, they provide dataset-specific validation references that illustrate how fuzzy equivalence relations can be used to define an expected performance benchmark before the output of a data-driven cyberattack detector is accepted as reliable.

The fuzzy-equivalence framework therefore serves a V&V role rather than functioning as an additional attack-detection algorithm. It provides an independent, data-structure-based reference for assessing the expected capability and reliability of anomaly-detection models [25].

2.3.2 GNSS Spoofing Onset Localization in Nonlinear State Estimators

The GNSS onset-localization research addresses post-alarm navigation integrity when a spoofing attack may have influenced the recursive state estimate before detection. Sequential integrity monitors generally require time to accumulate sufficient evidence before declaring an attack, creating a delay between the true spoofing onset and the alarm. Measurements received during this interval may already have corrupted the estimated navigation state. The proposed framework therefore retrospectively estimates the attack start time from the residual sequence of a nonlinear state estimator so that the portion of the recent navigation history potentially affected by spoofing can be identified [26].

Onset localization is formulated as an alarm-conditioned change-point estimation problem and examined through both statistical and learning-based approaches. The objectives address the statistical conditions under which spoofing onset can be inferred from estimator residuals, localization of the attack within the detection-delay interval, temporal correlation introduced by nonlinear state estimation, comparison of model-based and data-driven localizers, and the effects of attack strength and temporal evolution on localization accuracy [26].

2.3.2.1 Characterize GNSS Spoofing Onset Detectability

The analysis seeks to characterize the conditions under which the onset of a deceptive GNSS attack can be reliably inferred from a finite window of navigation-filter residuals. The normalized residual window is modeled using a Gaussian mean-shift formulation to evaluate the statistical separation produced by attack-induced changes in the estimator residuals [26].

Within this formulation, detectability is governed by a covariance-weighted residual spoof-energy measure. The analysis establishes a quantitative relationship among attack-induced residual energy, false-alarm probability, and achievable detection probability, thereby grounding onset localization in the statistical information contained in the estimator residuals rather than treating it solely as an empirical classification problem [26].

2.3.2.2 Localize the True Attack Onset Within the Detection-Delay Interval

The framework seeks to estimate the true spoofing onset within the interval preceding an integrity-monitor alarm. Rather than treating the alarm time as the attack start time, it searches retrospectively over an alarm-conditioned candidate window and identifies the residual change point most consistent with the observed post-onset behavior [26].

This distinction is important for navigation integrity because measurements received before the alarm may already have biased the state estimate. Localizing the onset provides a time reference for determining how much of the recent estimator history may have been influenced by spoofing and is therefore a prerequisite for future recovery, rollback, or reinitialization strategies that seek to restore a trustworthy navigation state [26].

2.3.2.3 Account for Temporal Correlation in Nonlinear Estimator Residuals

The onset-localization study aims to account explicitly for temporal dependence in the residual sequence generated by nonlinear state estimation. Classical change-point procedures are often derived under an independence assumption, whereas residuals from nonlinear filters may exhibit temporal correlation because of time-varying gains, recursive state propagation, and modeling approximations [26].

To address this dependence, the framework incorporates the covariance structure of the stacked residual window into the onset-localization formulation. This treatment leads to the correlated Generalized Likelihood Ratio (C-GLR) localizer, which weights candidate mean-shift signatures using the residual covariance rather than treating residual samples as independent and identically distributed [26].

2.3.2.4 Compare Model-Based and Data-Driven Onset Localization Methods

The onset-localization study aims to compare complementary model-based and data-driven strategies for estimating the spoofing onset. The evaluated methods include a retrospective Cumulative Sum (CUSUM) baseline, the proposed C-GLR localizer, and a temporal neural onset localizer that maps residual-derived features to a probability distribution over candidate onset indices [26].

This comparison is intended to determine the benefit obtained from progressively richer use of temporal information. CUSUM represents a conventional evidence-accumulation baseline; C-GLR explicitly models structured mean shifts and residual covariance; and the neural localizer learns temporal patterns directly from residual sequences. A second neural configuration further incorporates estimator diagnostics to evaluate whether internal filter information can improve localization precision beyond residual features alone [26].

2.3.2.5 Assess the Effects of Attack Energy and Attack Dynamics

The onset-localization study aims to assess how spoofing strength and temporal evolution affect localization performance. Particular attention is given to how covariance-weighted spoof energy and different attack profiles, including abrupt step biases and gradually increasing ramp biases, influence localization accuracy and uncertainty [26].

Contribution 2 therefore extends V&V beyond the binary question of whether spoofing has been detected. It evaluates whether the timing of the attack can be reconstructed from estimator evidence and how reliably that reconstruction can be performed under different statistical and attack-dynamic conditions [26].

2.4 Relationship Between the Two Subtasks

The two subtasks address complementary aspects of cybersecurity, verification, and resilience in HATS. **Subtask 1, Detection and Identification**, focuses on determining whether malicious or abnormal behavior is occurring, identifying compromised localization information, and supporting continued navigation when GNSS becomes unreliable. **Subtask 2, Data-Driven Verification and Validation of Attacked HATS**, extends this effort by assessing the expected reliability of data-driven anomaly-detection models and by retrospectively localizing the true onset of deceptive GNSS attacks within the interval between attack initiation and eventual detection.

The two subtasks provide a broader cybersecurity framework that addresses not only **whether an attack is occurring**, but also **how reliably the associated detection mechanisms can be evaluated and when the attack actually began**. This combination supports more trustworthy interpretation of detection outputs, improved assessment of model performance, and better identification of the period during which corrupted information may already have influenced the vehicle state. In this way, the project links attack detection with model validation and post-alarm onset localization to improve the safety, reliability, and cyber resilience of HATS.

3. Technical Approach and Methodology

3.1 Subtask 1: Detection and Identification

The Detection and Identification subtask was developed through a sequence of methodologies intended to progressively improve the capability of Highly Automated Transportation Systems (HATS) to recognize compromised localization information. The research began with consistency analysis between Global Navigation Satellite System (GNSS) measurements and vehicle motion independently derived from onboard sensors. This foundation was subsequently extended through data-driven displacement prediction, adaptive anomaly detection, time-series analysis, and, ultimately, investigation of navigation strategies that do not depend on GNSS after compromise or denial.

The methodologies described in this subtask were developed with particular attention to stealthy GNSS spoofing attacks. Unlike abrupt attacks that introduce immediately recognizable inconsistencies, sophisticated spoofing strategies can preserve physically plausible short-term vehicle motion while gradually manipulating the perceived vehicle state. Accordingly, the developed approaches progressively incorporate nonlinear motion modeling, statistical characterization of uncertainty, adaptive decision boundaries, and temporal information.

3.1.1 Contribution 1: Data-Driven GNSS Spoofing Detection

The first research contribution introduces a data-driven framework for detecting GNSS spoofing attacks using measurements available from onboard vehicle sensors. This contribution was published in the 2025 IEEE 4th International Conference on AI in Cybersecurity (ICAIC) [10]. The proposed framework combines vehicle speed and gyroscope information to estimate short-term vehicle displacement using a Deep Neural Network (DNN). The predicted displacement is continuously compared with the displacement and speed reported by GNSS to detect turn-by-turn,

stop, overshoot, and multiple small biased spoofing attacks. An analytical vehicle-motion model was also developed and evaluated as a baseline against which the data-driven approach could be compared.

A principal motivation for the data-driven formulation is the limitation introduced by analytical motion reconstruction. Analytical models provide interpretable estimates of vehicle motion but depend on explicit kinematic relationships and accumulate errors arising from sensor noise, integration, and modeling approximations. Consequently, relatively conservative thresholds may be required to avoid false alarms. The data-driven method instead learns the nonlinear relationship between onboard measurements and vehicle displacement directly from nominal driving data. By predicting displacement rather than absolute position, cumulative integration error is reduced and a tighter nominal error distribution can be obtained, thereby increasing sensitivity to smaller spoofing-induced discrepancies. This progression from analytical motion consistency to learned displacement prediction forms the methodological basis of Contribution 1.

3.1.1.1 Problem Formulation

An Automated Driving System (ADS) equipped with a wheel-speed sensor, an Inertial Measurement Unit (IMU), and a GNSS receiver is considered. The vehicle operates over discrete time steps indexed by k, with a sampling interval Δt. The threat model assumes that the attacker is capable of manipulating the GNSS information received by the vehicle while the onboard wheel-speed and inertial measurements remain uncompromised. The onboard measurements can therefore provide an independent representation of physical vehicle motion against which the externally supplied GNSS information can be evaluated.

At each time step, the onboard motion measurements are represented by

$$\mathbf{s}_k = [v_k, \psi_k, \Delta t_k] \tag{3.1}$$

where v_k denotes vehicle speed obtained from the wheel-speed sensor, ψ_k denotes the vehicle heading or yaw angle derived from inertial measurements, and Δt_k represents the corresponding time interval.

A finite temporal window of recent onboard measurements is provided to a data-driven displacement predictor. The resulting planar displacement estimate is expressed as

$$\hat{\mathbf{d}}_k = \left[\hat{d}_{x,k}, \hat{d}_{y,k}\right] \tag{3.2}$$

Because this displacement is predicted exclusively from onboard sensor information, it is independent of the GNSS signal being evaluated. In parallel, GNSS position measurements are used to determine GNSS-derived planar displacement and speed,

$$\mathbf{d}_k^{\text{GNSS}} = \left[d_{x,k}^{\text{GNSS}}, d_{y,k}^{\text{GNSS}}\right], \quad v_k^{\text{GNSS}} \tag{3.3}$$

The fundamental detection problem is then expressed as a motion-consistency comparison between the displacement predicted from trusted onboard sensing and the displacement perceived through GNSS. The displacement residual is defined as

$$\mathbf{r}_k = \hat{\mathbf{d}}_k - \mathbf{d}_k^{\text{GNSS}} \tag{3.4}$$

Under attack-free driving conditions, the difference between these two motion estimates reflects sensor noise, model uncertainty, and normal variations in vehicle operation. These discrepancies remain statistically bounded. During GNSS spoofing, the externally reported motion progressively becomes inconsistent with the physical motion measured by the vehicle, causing a corresponding change in the residual behavior. The detection objective is therefore to distinguish spoofing-induced discrepancies from the residual variations expected during nominal operation.

Four attack scenarios are considered. Turn-by-turn attacks manipulate the vehicle trajectory perceived from GNSS and may cause incorrect routing while preserving smooth motion characteristics. Stop attacks cause a stationary vehicle to perceive motion through manipulated GNSS information. Overshoot attacks represent the inverse condition, in which a moving vehicle is falsely perceived by GNSS as stationary or moving substantially slower than its actual motion. The fourth and more challenging scenario consists of multiple small biased attacks, in which each individual manipulation is sufficiently small to remain below an instantaneous detection threshold but repeated manipulations gradually produce a significant localization deviation.

The multiple small biased scenario is particularly important because it demonstrates that detection based only on the magnitude of individual measurements may be insufficient. An attacker may preserve short-term physical plausibility while gradually altering the long-term vehicle state. Consequently, the methodology was designed not only to detect conventional large-magnitude attacks, but also to investigate whether the tighter uncertainty bounds provided by learned displacement prediction can expose smaller manipulations.

3.1.1.2 In-Vehicle Sensor Measurements

The research uses the Honda Research Institute Driving Dataset (HDD) [14] as its primary real-world data source. HDD contains approximately 104 hours of naturalistic driving data collected across 137 driving sessions in the San Francisco Bay Area. The data include urban streets, suburban roads, and highways and contain natural vehicle behaviors including stopping, turning, lane changing, acceleration, and deceleration. These diverse operating conditions provide the variation necessary for evaluating motion-consistency-based spoofing detection beyond a controlled or scripted driving environment.

HDD provides synchronized information from several sensing modalities. The localization information is obtained from an Automotive Dynamic Motion Analyzer (ADMA) integrated with differential Global Positioning System (DGPS) measurements. The available information includes latitude, longitude, and heading. The IMU provides angular velocity about the vehicle axes as well as linear acceleration. The Controller Area Network (CAN) bus provides vehicle-dynamics information including vehicle speed, steering-wheel angle, steering-wheel rotational speed, brake pressure, accelerator-pedal position, and turn-signal information. The primary signals used for the spoofing-detection methodology are the GNSS measurements, vehicle speed, and yaw-related inertial information.

The principal sensor streams are originally available at a sampling rate of 100 Hz. Although this high temporal resolution captures fine-scale vehicle dynamics, processing every measurement increases computational demand and introduces considerable temporal redundancy. The sensor streams were therefore down-sampled by selecting one measurement from every ten samples, resulting in an effective processing frequency of 10 Hz.

This sampling strategy represents a compromise between temporal resolution and computational efficiency. At approximately 70 mph, or 31.3 m/s, a 10-Hz update rate corresponds to approximately 3.2 m of vehicle travel between successive processed samples. The resulting spatial resolution remains sufficiently fine for the vehicle-motion events considered while reducing the number of computations required for online processing. Down-sampling also reduces the influence of high-frequency sensor fluctuations and facilitates construction of fixed-duration temporal input windows.

The camera and LiDAR measurements available in HDD are not required for this contribution. The spoofing-detection architecture relies on the relationship between GNSS and trusted onboard motion measurements. Camera-based navigation is addressed separately in Contribution 4 of this report.

3.1.1.3 Analytical Detection Baseline

An analytical motion-consistency framework was developed as a baseline before introducing the learned displacement model. The analytical framework independently reconstructs vehicle motion using the speedometer and gyroscope and compares this estimate with motion inferred from GNSS. Because the reconstructed trajectory does not depend on the GNSS measurements, it provides an independent reference for identifying inconsistencies introduced by spoofing.

The gyroscope measures vehicle angular velocity about the vertical axis. The yaw-rate measurement is numerically integrated to estimate vehicle heading. Let $\psi(t_k)$ denote yaw rate at processed time instant t_k. The heading is updated according to

$$\psi(t_k) = \psi(t_{k-1}) + \dot{\psi}(t_k)\Delta t \tag{3.5}$$

The speedometer provides the magnitude of the vehicle velocity. Using the estimated heading, the vehicle velocity is decomposed into planar components as

$$v_x(t_k) = v(t_k)\cos\big(\psi(t_k)\big) \tag{3.6}$$

$$v_y(t_k) = v(t_k)\sin\big(\psi(t_k)\big) \tag{3.7}$$

Incremental displacement over one processed time interval is consequently calculated as

$$\Delta x(t_k) = v_x(t_k)\Delta t \tag{3.8}$$

$$\Delta y(t_k) = v_y(t_k)\Delta t \tag{3.9}$$

The local vehicle position is then recursively reconstructed using

$$x(t_k) = x(t_{k-1}) + \Delta x(t_k) \tag{3.10}$$

$$y(t_k) = y(t_{k-1}) + \Delta y(t_k) \tag{3.11}$$

These relationships produce a trajectory based only on onboard physical-motion measurements. The reconstruction remains independent of GNSS spoofing, although sensor noise and integration error can introduce increasing uncertainty over time.

GNSS provides a second and independently calculated representation of vehicle motion. Geographic position is represented by latitude and longitude. Given consecutive GNSS coordinates (ϕ_{k-1}, λ_{k-1}) and (ϕ_k, λ_k), the distance between the two points is calculated using the Haversine formulation [15]:

$$d_{\text{GNSS}}(t_k) = 2R\sin^{-1}\left(\sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + \cos(\phi_{k-1})\cos(\phi_k)\sin^2\left(\frac{\Delta\lambda}{2}\right)}\right) \tag{3.12}$$

where $\Delta\phi = \phi_k - \phi_{k-1}$, $\Delta\lambda = \lambda_k - \lambda_{k-1}$, and R = 6,378,000 m is the Earth radius used in the analytical implementation. The GNSS-derived average speed between consecutive processed measurements is then

$$v_{\text{GNSS}}(t_k) = \frac{d_{\text{GNSS}}(t_k)}{\Delta t} \tag{3.13}$$

Under nominal conditions, the GNSS-derived displacement and speed are expected to remain consistent with the physical motion measured onboard. A spoofing attack alters this relationship.

Let ($X^{IMU}(k)$, $Y^{IMU}(k)$) represent the locally reconstructed vehicle position and ($X^{GNss}(k)$, $Y^{GNss}(k)$) represent the corresponding GNSS position. The longitudinal and lateral position residuals are

$$r_x(k) = X_{\text{IMU}}(k) - X_{\text{GNSS}}(k) \tag{3.14}$$

$$r_y(k) = Y_{\text{IMU}}(k) - Y_{\text{GNSS}}(k) \tag{3.15}$$

The Euclidean position residual is

$$r_{\text{pos}}(k) = \sqrt{r_x^2(k) + r_y^2(k)} \tag{3.16}$$

Similarly, the speed residual is defined as

$$r_v(k) = v_{\text{IMU}}(k) - v_{\text{GNSS}}(k) \tag{3.17}$$

Thresholds are extracted from clean data in which no GNSS manipulation is present. If D_{clean} denotes the clean dataset, the position threshold is

$$\delta_{\text{pos}} = \max_{k \in \mathcal{D}_{\text{clean}}} r_{\text{pos}}(k) \tag{3.18}$$

and the speed threshold is

$$\delta_v = \max_{k \in \mathcal{D}_{\text{clean}}} |r_v(k)| \tag{3.19}$$

These thresholds represent conservative upper bounds on the discrepancies observed during nominal operation. A threshold violation indicates motion inconsistency and initiates the attack-classification process.

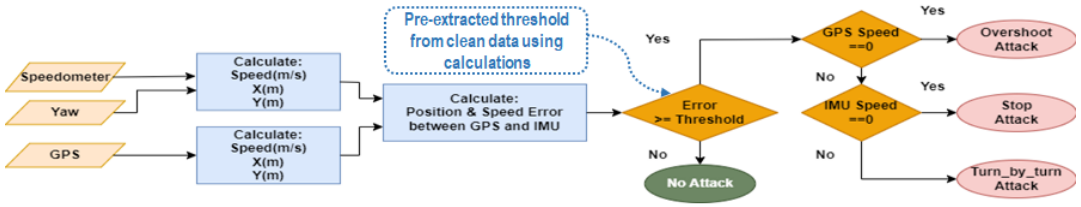


Figure 3.1. Analytical GNSS spoofing detection framework. Vehicle motion is reconstructed independently from in-vehicle sensors and GNSS, and residuals are evaluated against thresholds extracted from clean data. Conditional logic based on GNSS and in-vehicle speed enables classification of turn-by-turn, stop, and overshoot spoofing attacks.

The analytical architecture contains two parallel motion-estimation paths, as illustrated in Figure 3.1. The first estimates motion from the speedometer and gyroscope, whereas the second calculates motion from GNSS. Residuals between the two paths are evaluated against thresholds extracted from clean data. When the residuals remain within nominal bounds, the GNSS information is treated as consistent with physical vehicle motion. When a threshold is exceeded, vehicle-speed relationships are used to distinguish among attack behaviors.

The analytical baseline is computationally simple and provides interpretable detection decisions. However, its dependence on integrated vehicle motion produces accumulated estimation error. The maximum nominal error must therefore be incorporated into the threshold, which can reduce sensitivity to small GNSS manipulations. This limitation motivated the displacement-based data-driven formulation described next.

3.1.1.4 Data-Driven Displacement Prediction

The data-driven framework retains the principle of motion consistency while replacing analytical position reconstruction with learned displacement prediction. Instead of repeatedly integrating vehicle velocity to construct a complete trajectory, short-horizon displacement is predicted directly from a finite window of trusted onboard measurements. The predicted motion is then compared with displacement derived from GNSS.

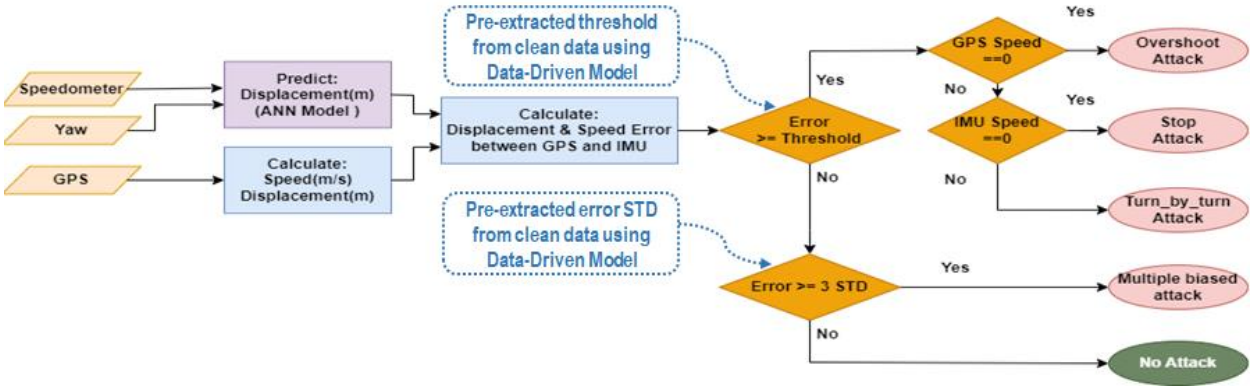


Figure 3.2. Data-driven GNSS spoofing attack detection framework. In-vehicle sensor measurements are processed by a neural-network displacement predictor and compared against GNSS-derived motion to compute residuals for spoofing detection and classification.

As shown in Figure 3.2, the methodology contains three primary stages: sensor preprocessing and feature construction, data-driven displacement prediction, and residual-based attack detection and classification. Measurements are synchronized and down-sampled before being organized into a sliding temporal window. Vehicle speed and yaw information are provided to the DNN, while GNSS measurements are retained outside the network and are used to calculate the displacement and speed against which the DNN output is compared.

Direct use of heading angle as an input creates an artificial discontinuity when the angle wraps between +π and −π. To avoid this discontinuity, heading is represented using its sine and cosine components:

$$[\sin(\psi_k), \cos(\psi_k)] \tag{3.20}$$

This transformation preserves the directional information while providing bounded and continuous features for the learning process.

Vehicle displacement depends not only on an instantaneous measurement but also on the recent motion history of the vehicle. A fixed temporal window is therefore constructed from consecutive processed samples. Let N denote the window length. At time k, the DNN input is

$$\mathbf{X}_k = \begin{bmatrix} v_{k-N+1}, \dots, v_k \\ \sin(\psi_{k-N+1}), \dots, \sin(\psi_k) \\ \cos(\psi_{k-N+1}), \dots, \cos(\psi_k) \\ \Delta t_{k-N+1}, \dots, \Delta t_k \end{bmatrix} \tag{3.21}$$

A window containing ten consecutive processed samples is used. Because the data are processed at 10 Hz, each window spans approximately one second of vehicle motion. Four quantities are provided for each of the ten samples: vehicle speed, sine of heading, cosine of heading, and time interval. The resulting DNN input therefore contains 40 features.

The learning target is defined as relative displacement over the temporal window rather than absolute position. Given vehicle position (x_k, y_k), the target corresponding to X_k is

$$\mathbf{d}_k = \begin{bmatrix} x_k - x_{k-N} \\ y_k - y_{k-N} \end{bmatrix} \tag{3.22}$$

This formulation is important because each prediction is independent of earlier predicted displacements. The output of one window does not need to be integrated into the next network prediction. Consequently, prediction error does not accumulate through the learning architecture in the same manner as analytical position reconstruction. The independent-window formulation also permits the use of a feedforward network rather than a recurrent architecture.

The DNN produces two outputs corresponding to planar displacement:

$$\hat{\mathbf{d}}_k = [\hat{d}_{x,k}, \hat{d}_{y,k}] \tag{3.23}$$

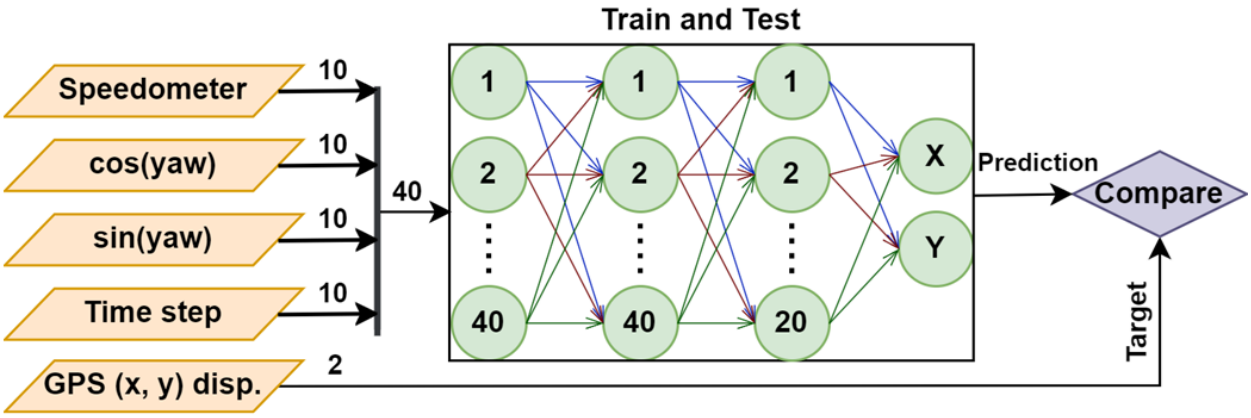


Figure 3.3. Neural-network architecture for data-driven vehicle displacement prediction. A temporal window of in-vehicle sensor features is processed through fully connected layers to predict planar displacement components.

The input vector contains the 40 features described above. The network consists of two 40-neuron layers followed by a 20-neuron layer and a two-neuron output layer. Hyperbolic tangent activation functions are applied to the hidden layers, while the output layer uses a linear activation function. The nonlinear hidden-layer activations allow interactions among vehicle speed, heading, and temporal behavior to be modeled, while the linear output avoids restricting the range of continuous displacement predictions. The architecture used in the ICAIC implementation follows the 40-40-20-2 configuration shown in Figure 3.3.

Training is performed by minimizing the Mean Absolute Error (MAE) between predicted and target displacement. For M training samples, the objective is

$$\mathcal{L} = \frac{1}{M}\sum_{k=1}^{M} \|\hat{\mathbf{d}}_k - \mathbf{d}_k\|_1 \tag{3.24}$$

where $\|\cdot\|_1$ denotes the $\ell 1$ norm. MAE provides a linear penalty for prediction error and reduces the excessive influence that isolated large errors can have during optimization. Network parameters are optimized using the Adam optimizer with mini-batch training.

The DNN development dataset contains 144,000 samples. Seventy percent, corresponding to 100,800 samples, is used for training, while the remaining 43,200 samples are used for testing. The model is trained for 100 epochs using a batch size of 32. The network configuration is summarized in Table 3.1.

Table 3.1. Hyperparameters of the developed data-driven displacement prediction model.

Hyperparameter	Value
Number of neurons, first layer	40

Hyperparameter	Value
Number of neurons, second layer	40
Number of neurons, third layer	20
Number of neurons, output layer	2
Number of training epochs	100
Batch size	32
Activation function, hidden layers	tanh
Activation function, output layer	linear
Loss function	Mean Absolute Error
Optimizer	Adam

The training and testing MAE values are monitored throughout the optimization process. Training is terminated after 100 epochs because further epochs do not provide a meaningful reduction in loss. The close behavior of the training and testing curves is used to evaluate convergence and the ability of the DNN to generalize to samples not used for parameter optimization.



Figure 3.4. Training and testing loss curves for the data-driven displacement prediction model using Mean Absolute Error loss. The smooth convergence and close alignment of the curves indicate stable training and generalization performance.

Once trained, the DNN parameters remain fixed during attack detection. GNSS information is not used as a network input during online displacement prediction. It is instead retained as the potentially compromised information source against which the independent DNN prediction is evaluated.

3.1.1.5 Attack Detection Logic

The detection stage quantifies the disagreement between DNN-predicted vehicle motion and GNSS-derived vehicle motion. Two complementary residuals are considered: a displacement residual and a speed residual.

Let $\hat{d}_k$ represent displacement predicted by the DNN and $d^{GNss}{}_k$ represent displacement derived from GNSS. The displacement residual vector is

$$\mathbf{r}_k^{\text{disp}} = \hat{\mathbf{d}}_k - \mathbf{d}_k^{\text{GNSS}} \tag{3.25}$$

For comparison with a scalar threshold, its Euclidean magnitude is calculated as

$$r_k^{\text{disp}} = \left\|\mathbf{r}_k^{\text{disp}}\right\|_2 \tag{3.26}$$

The speed residual is calculated using the difference between onboard vehicle speed and GNSS-derived speed:

$$r_k^{\text{speed}} = \left|\hat{v}_k - v_k^{\text{GNSS}}\right| \tag{3.27}$$

The displacement residual identifies spatial inconsistency between the two motion estimates. The speed residual provides complementary information that is particularly useful for distinguishing stop and overshoot attacks.

Clean data are processed through the trained DNN to characterize nominal residual behavior. The mean and standard deviation of the displacement residual are calculated as

$$\mu_{\text{disp}} = \frac{1}{N}\sum_{k=1}^{N} r_k^{\text{disp}} \tag{3.28}$$

$$\sigma_{\text{disp}} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(r_k^{\text{disp}} - \mu_{\text{disp}}\right)^2} \tag{3.29}$$

Similarly, the speed residual statistics are

$$\mu_{\text{speed}} = \frac{1}{N}\sum_{k=1}^{N} r_k^{\text{speed}} \tag{3.30}$$

$$\sigma_{\text{speed}} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(r_k^{\text{speed}} - \mu_{\text{speed}}\right)^2} \tag{3.31}$$

For conventional large-magnitude attacks, the maximum residual observed in clean data is used as the corresponding detection threshold. This procedure incorporates the largest nominal discrepancy arising from sensor noise and prediction uncertainty. When either displacement or speed residual exceeds its corresponding bound, an anomaly is declared.

Attack identification is subsequently performed using the relationship between the two residuals and vehicle-speed information. A turn-by-turn attack is characterized by a significant displacement inconsistency while both GNSS-derived and onboard speeds continue to indicate forward motion. The detection condition is

$$r_k^{\text{disp}} > \delta_{\text{disp}} \quad \text{and} \quad \hat{v}_k > \delta_{\text{speed}} \quad \text{and} \quad v_k^{\text{GNSS}} > \delta_{\text{speed}} \tag{3.32}$$

A stop spoofing attack occurs when GNSS indicates motion while the onboard vehicle measurements indicate that the vehicle is stationary. The corresponding condition is

$$r_k^{\text{speed}} > \delta_{\text{speed}} \quad \text{and} \quad \hat{v}_k \leq \delta_{\text{speed}} \tag{3.33}$$

An overshoot attack represents the opposite inconsistency. The physical vehicle remains in motion while GNSS indicates a stationary or near-stationary state:

$$r_k^{\text{speed}} > \delta_{\text{speed}} \quad \text{and} \quad v_k^{\text{GNSS}} \leq \delta_{\text{speed}} \tag{3.34}$$

The explicit decision logic retains the interpretability of the analytical baseline. The primary difference is that the displacement against which GNSS is compared is no longer reconstructed through accumulated analytical integration. Instead, it is predicted independently over a short horizon by the trained DNN.

3.1.1.6 Multiple Small Biased Attack Detection

The multiple small biased attack scenario is designed to evaluate the detector under a more stealthy form of GNSS manipulation. In this scenario, each individual spoofing step is intentionally maintained below the conventional maximum-residual threshold. If each measurement is considered independently, no single manipulation necessarily produces sufficient error to trigger the conventional detector. Repeated manipulations can nevertheless create a substantial accumulated localization deviation.

The data-driven displacement model provides an advantage for this condition because its nominal prediction errors exhibit tighter statistical bounds than those produced by accumulated analytical position reconstruction. Rather than using only the maximum nominal error, the distribution of clean displacement errors is characterized statistically. Based on the Gaussian-error treatment adopted

in the ICAIC work, approximately 99.73% of nominal observations are expected to fall within three standard deviations of the mean [10].

The statistical threshold used for multiple small biased attack detection is therefore

$$\delta_{\text{disp}}^{\text{msb}} = \mu_{\text{disp}} + 3\sigma_{\text{disp}} \tag{3.35}$$

where μ^{disp} and σ^{disp} are calculated from the clean displacement residual distribution.

The attack-generation procedure illustrates the distinction between individual attack magnitude and cumulative localization consequence. The GNSS coordinates are manipulated by 1 m along both planar coordinate directions. The Euclidean displacement introduced by an individual manipulation is therefore approximately

$$d_{\text{attack}} = \sqrt{1^2 + 1^2} \approx 1.41 \text{ m} \tag{3.36}$$

A sequence of 101 such small biased manipulations produces an accumulated location deviation of approximately 141 m, even though each approximately 1.41-m individual manipulation remains below the conventional system threshold. The ICAIC experiment was explicitly designed to test this condition.

This attack model demonstrates a fundamental limitation of instantaneous threshold monitoring. A small discrepancy may be statistically plausible at a single time step but become highly consequential when introduced repeatedly. The data-driven framework improves sensitivity by reducing the nominal uncertainty boundary, although persistent temporal attack behavior is addressed more explicitly in Contribution 3 through time-series analysis.

3.1.1.7 Dataset and Attack Generation

The experimental methodology combines real-world benign driving data with controlled synthetic GNSS spoofing injection. HDD [14] is used because it provides long-duration, synchronized GNSS, IMU, CAN, camera, and LiDAR data collected under natural driving conditions. The detection framework itself primarily uses vehicle speed, yaw-related inertial information, and GNSS measurements. HDD records these signals at 100 Hz before the 10-Hz processing strategy described previously is applied.

Because the original HDD sequences do not contain naturally occurring GNSS spoofing events, attacked datasets are generated by modifying GNSS measurements offline while maintaining the corresponding onboard measurements unchanged. This construction preserves the real vehicle dynamics and natural sensor-noise characteristics of the recorded sequence while allowing the exact attack interval and manipulation magnitude to be controlled. Each timestamp can consequently be assigned a known normal or attack label for quantitative evaluation.

Separate attack scenarios are generated for turn-by-turn, stop, overshoot, and multiple small biased attacks. The injected GNSS trajectories are constructed to preserve realistic motion continuity rather than introducing obvious discontinuities that could be detected through simple signal-quality checks. Attack parameters are varied across driving segments containing different vehicle speeds, road geometries, and operating conditions. The in-vehicle measurements remain unmodified during

spoofing injection so that they continue to represent the physical vehicle motion used as the independent reference.

For development of the DNN, 144,000 nominal samples are used, with 100,800 assigned to training and 43,200 assigned to testing. The trained model is then used to calculate displacement residuals from clean and attacked sequences. Clean residuals provide the conventional error thresholds as well as the statistical parameters used by the 3σ detector. Attacked sequences are subsequently processed without retraining the DNN.

The resulting methodology establishes a controlled comparison between the analytical and data-driven detection approaches. Both methods operate on the same underlying real-world vehicle behavior, while the GNSS component is modified according to predefined spoofing scenarios. Consequently, differences in detection capability can be attributed to the motion-estimation and detection formulations rather than to differences in the underlying driving data.

The quantitative outcomes of this evaluation, including the detection performance for turn-by-turn, stop, overshoot, and multiple small biased attacks, are presented separately in Section 4.1.1 of this report. Keeping these numerical results in Chapter 4 maintains the required separation between the Technical Approach and Methodology and Results and Findings sections of the CARMEN+ final-report template.

3.1.2 Contribution 2: Adaptive GNSS Spoofing Detection Using DBSCAN

The second research contribution extends the displacement-based GNSS spoofing detection framework by replacing a purely static anomaly boundary with an adaptive density-based mechanism. This contribution was published in the 2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC) as “GPS Spoofing Attack Detection in Autonomous Vehicles Using Adaptive DBSCAN” [11]. The objective is to preserve the strong detection capability of the data-driven displacement framework for conventional spoofing attacks while improving robustness when the nominal displacement-error distribution changes over time and when attacks are deliberately introduced through small incremental biases.

The methodology builds directly on Contribution 1. The same onboard sensing concept is retained: vehicle motion predicted from speed and inertial information is compared with GNSS-reported motion, and the resulting displacement error is used as the primary indicator of inconsistency. The principal modification is the introduction of Density-Based Spatial Clustering of Applications with Noise (DBSCAN) as an online anomaly detector whose neighborhood radius is updated from recursively estimated statistics of non-anomalous displacement errors. In this manner, the detection boundary is allowed to evolve with the operating condition rather than remaining fixed for the duration of vehicle operation.

3.1.2.1 Limitations of Static Detection Thresholds

The data-driven framework described in Section 3.1.1 improves spoofing detection by producing substantially tighter nominal displacement errors than the analytical position-reconstruction baseline. Nevertheless, its conventional decision logic still depends on thresholds extracted from attack-free data. A fixed threshold implicitly assumes that the statistical behavior of the nominal

residual remains sufficiently stationary after calibration. In real driving environments, this assumption may not always hold because vehicle dynamics, road geometry, maneuver intensity, sensor noise, and environmental conditions can change over time.

A static boundary creates two competing risks. If the threshold is selected too tightly, a shift in the nominal residual distribution can cause normal samples to cross the boundary, thereby increasing false alarms. If the threshold is selected too conservatively, the detector becomes less sensitive to subtle spoofing-induced deviations. This tradeoff becomes especially important for multiple small biased attacks, where individual GNSS manipulations are intentionally kept below the conventional system threshold while their accumulated effect produces a large localization error.

The limitation is not restricted to conventional scalar thresholding. Standard DBSCAN also depends strongly on the neighborhood radius ε. A fixed ε assumes that the density of nominal samples is approximately stationary. If ε is too small, legitimate variations in the displacement-error distribution can be labeled as noise. If ε is too large, spoofing-induced samples can remain connected to the nominal cluster and therefore escape detection. The adaptive framework was developed to make ε responsive to the current statistical uncertainty of the nominal displacement residual rather than treating it as a fixed hyperparameter.

3.1.2.2 Adaptive DBSCAN Framework

The proposed framework reformulates the detection of subtle GNSS spoofing as a density-based anomaly-detection problem. The displacement predictor developed in Contribution 1 is retained as the nominal motion model. No change to the basic sensor inputs is required: speed and inertial measurements are used to predict short-horizon vehicle displacement, while GNSS measurements are processed independently to obtain the potentially compromised displacement. The scalar displacement residual already defined in Eq. (3.26) is then supplied to the adaptive detection stage.

The overall decision structure contains two complementary branches. When the displacement or speed error exceeds the pre-extracted conventional thresholds, the rule-based logic described in Section 3.1.1 is used to detect and classify turn-by-turn, stop, and overshoot attacks. When the displacement error remains below those static bounds, the adaptive DBSCAN module evaluates whether the sample remains consistent with the current density of nominal displacement errors. This second branch is intended primarily to expose gradual and multiple small biased attacks that can remain hidden inside a conventional static threshold.

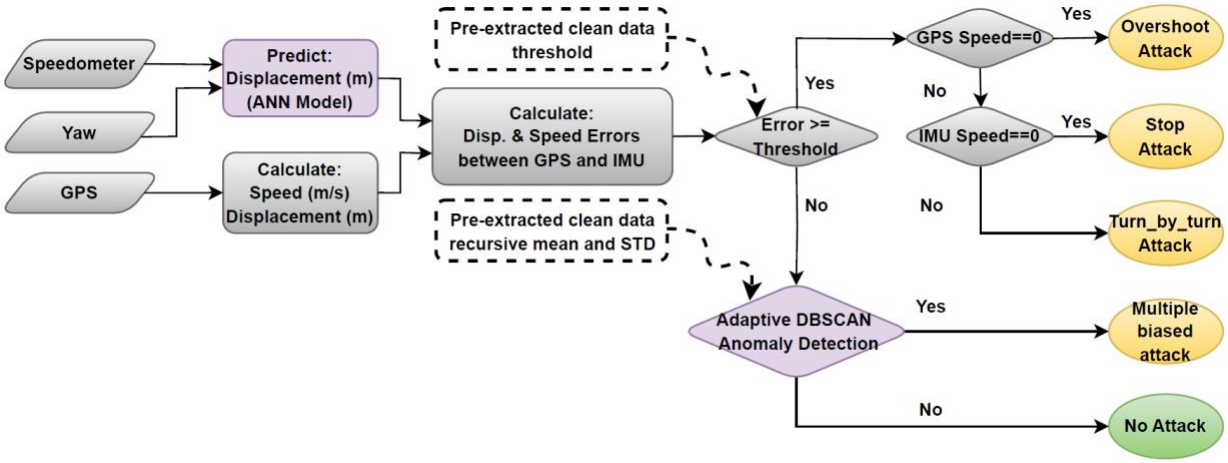


Figure 3.5. Adaptive DBSCAN-based GNSS spoofing detection framework. Conventional large-magnitude attacks are identified using pre-extracted displacement and speed thresholds, while multiple small biased attacks are evaluated using an adaptive DBSCAN module whose neighborhood radius is updated from recursively estimated clean-data statistics.

DBSCAN identifies dense regions of samples without requiring a predefined number of clusters. In its conventional form, a point is considered a core point when a sufficient number of neighboring samples lie within an ε-neighborhood; border points lie within the neighborhood of a core point, and samples that satisfy neither condition are treated as noise. For spoofing detection, nominal displacement errors are expected to occupy the dominant dense region, whereas errors that depart from this structure are interpreted as potential attack samples.

The adaptive implementation used in this work is simplified for online anomaly detection. The modified detector maintains a single primary cluster representing nominal behavior. The SMC implementation does not rely on a separate minimum-points requirement for the anomaly decision; a displacement-error sample lying outside the current ε boundary is treated as anomalous regardless of the number of neighboring anomalous samples [11]. Most importantly, ε is recalculated as new non-anomalous samples are accepted, allowing the density boundary to track gradual changes in the nominal error distribution.

3.1.2.3 Recursive Mean and Standard-Deviation Estimation

Adaptation is achieved through recursive estimation of the mean and dispersion of the displacement error. Let e_k denote the incoming displacement residual at time step k and let μ_k denote the estimated mean of the nominal residual distribution. Rather than storing the complete history and recomputing the mean after every sample, the estimate is updated incrementally using only samples that have been classified as non-anomalous:

$$\mu_k = \mu_{k-1} + \frac{1}{n_k}(e_k - \mu_{k-1}) \tag{3.37}$$

where n_k denotes the number of samples accepted as non-anomalous up to the current update. The recursive formulation allows gradual changes in the nominal displacement error to be tracked using constant memory. Because the update is performed only after the current sample has passed the anomaly test, a detected spoofing sample does not directly shift the estimated nominal mean.

The uncertainty of the nominal displacement error is updated in the same manner. Let σ^2_k denote the estimated variance at time step k. The recursive variance update used in the dissertation formulation is

$$\sigma_k^2 = \sigma_{k-1}^2 + \frac{1}{n_k}\left[(e_k - \mu_{k-1})^2 - \sigma_{k-1}^2\right] \tag{3.38}$$

with the corresponding standard deviation given by $\sigma_k = \sqrt{(\sigma^2_k)}$. The same selective-update rule is applied to this estimate: only non-anomalous displacement errors are allowed to modify the variance. Consequently, the nominal uncertainty can evolve smoothly as normal driving conditions change while abrupt or spoofing-induced outliers are prevented from inflating the statistical boundary.

Recursive estimation has two practical advantages for an onboard detector. First, the number of arithmetic operations required at each sample does not grow with the length of the driving sequence. Second, the method does not require storing a long history of displacement errors. These properties are important for maintaining low memory usage and predictable computational demand during continuous vehicle operation.

3.1.2.4 Dynamic DBSCAN Radius ε

The defining feature of the adaptive detector is the dynamic neighborhood radius. Instead of selecting ε once during offline calibration, the radius is tied directly to the recursively estimated standard deviation of the current nominal displacement-error distribution:

$$\varepsilon_k = 5\sigma_k \tag{3.39}$$

A three-standard-deviation bound contains approximately 99.73% of samples under a Gaussian assumption. The adaptive DBSCAN framework intentionally uses the more conservative 5σ radius. This choice provides additional tolerance for nominal uncertainty and for residual deviations associated with previously missed anomalies, while the selective-update mechanism prevents samples classified as anomalous from contaminating the nominal statistics. The objective is therefore not simply to reduce ε, but to maintain a boundary that follows legitimate distribution changes without drifting toward spoofed behavior.

The use of recursive statistics also changes the interpretation of ε from a static tuning parameter into an online estimate of the current uncertainty level. If normal displacement errors become more dispersed because of changing operating conditions, σ_k and therefore ε_k can increase. If the nominal error distribution becomes tighter, the radius can decrease. In both cases, the decision boundary is derived from samples that have already been accepted as nominal.

Algorithm 3.1. Adaptive DBSCAN with Recursive Statistics

	Require: Initial mean μ_0, initial standard deviation σ_0
1	Initialize $n \leftarrow 1, \mu \leftarrow \mu_0, \sigma \leftarrow \sigma_0$
2	while new displacement error e_k is available do
3	$\varepsilon \leftarrow 5\sigma$
4	if $\lvert e_k - \mu \rvert \leq \varepsilon$ then
5	$n \leftarrow n + 1$
6	$\mu \leftarrow \mu + (e_k - \mu)/n$
7	$\sigma^2 \leftarrow \sigma^2 + \left((e_k - \mu)^2 - \sigma^2\right)/n$
8	$\sigma \leftarrow \sqrt{\sigma^2}$
9	anomaly ← False
10	else
11	anomaly ← True
12	end if
13	end while

3.1.2.5 Initial Clean-Data Calibration

Although the detector adapts online, reliable initialization is required before recursive updates can begin. A total of 120,000 clean, non-spoofed displacement-residual samples are used to compute the initial mean μ_0 and standard deviation σ_0 of the nominal error distribution. These samples are used only for initialization of the statistical model; the adaptive stage subsequently updates the statistics from incoming data.

The use of a large attack-free initialization set is intended to provide a stable representation of normal displacement-prediction uncertainty at the start of operation. After initialization, the detector does not repeatedly process the 120,000 historical samples. Instead, each newly accepted nominal residual contributes one recursive update. This distinction is important for real-time feasibility because the computational effort during operation is independent of the size of the initialization dataset.

The initialization procedure also separates nominal-model formation from attack evaluation. Spoofed samples are not included in the initial estimates, and samples flagged as anomalous during online operation are excluded from later updates. This prevents the decision boundary from being progressively pulled toward attack-generated residuals, a failure mode that would otherwise reduce sensitivity during a prolonged or slowly evolving spoofing sequence.

3.1.2.6 Online Anomaly Detection

After initialization, displacement residuals are processed sequentially as they become available. For each sample, the current mean and standard deviation define the adaptive ε through Eq. (3.39). The incoming residual is then evaluated relative to the nominal cluster. If the sample lies within the current neighborhood boundary, it is labeled non-anomalous and is used to update μ and σ through Eqs. (3.37) and (3.38). If the sample lies outside the boundary, it is immediately labeled anomalous and is excluded from both recursive updates.

This selective-update logic is central to the stability of the framework. If spoofed measurements were allowed to update the recursive statistics, a gradual attack could progressively move the estimated mean or enlarge the standard deviation until the malicious behavior itself became incorporated into the nominal model. Excluding anomalous samples prevents this form of threshold drift and preserves the detector's representation of attack-free behavior.

The adaptive DBSCAN stage is executed only after the conventional threshold branch has been evaluated. Accordingly, large or structured spoofing events that already violate the displacement or speed thresholds continue to be handled by the established rule-based classification logic, while below-threshold displacement deviations are passed to the adaptive density test. The resulting hybrid architecture preserves the interpretability of the earlier detector while adding sensitivity to changes in the statistical structure of the residuals.

The online implementation is computationally lightweight. Recursive mean and variance estimation require a constant number of operations for each accepted sample, and no sliding-window reconstruction or batch re-clustering of the complete historical dataset is required. The displacement residual is processed once as it arrives, permitting low-latency operation and constant memory usage. These characteristics support implementation on onboard computing platforms where processing time and memory availability are constrained.

3.1.2.7 Dataset and Spoofing Scenarios

The adaptive DBSCAN framework is evaluated using the Honda Research Institute Driving Dataset (HDD) [14], the same real-world data source used for Contribution 1. The dataset provides synchronized GNSS, inertial, and vehicle-motion measurements from urban, suburban, and highway driving. The preprocessing procedure remains consistent with the preceding contribution, including synchronization of the relevant signals, calculation of GNSS and predicted displacement, and construction of the displacement residual supplied to the anomaly detector.

The data-driven displacement predictor is retained as the underlying nominal-motion estimator so that the effect of the adaptive detection mechanism can be evaluated independently of changes to the prediction model. In the dissertation formulation, no architectural modification or retraining is introduced for the adaptive DBSCAN experiments. The principal experimental change is therefore the replacement of the fixed below-threshold decision boundary by the recursively updated density-based detector.

Five clean HDD subsets are selected to represent different real-world driving conditions and speed profiles. The same four spoofing categories used in Contribution 1 are considered: turn-by-turn, stop, overshoot, and multiple small biased attacks. For conventional attack types, randomized attack magnitudes, durations, and start and end times are used across the selected subsets. Each attack category is evaluated through 25 independent attack instances, providing repeated trials across different vehicle-motion conditions.

The multiple small biased scenario is generated by incrementally manipulating the GNSS planar coordinates. The x and y coordinates are each shifted by 1 m per attack step for 101 consecutive steps. Each individual step therefore produces an approximately 1.41-m Euclidean displacement change, while the repeated sequence accumulates to approximately 141 m of localization deviation. The experiment is repeated five times on each of the five selected clean subsets, producing 25

multiple-small-biased attack realizations. The purpose of this design is to evaluate whether the adaptive detector can identify a sequence whose individual steps remain below the conventional displacement threshold.

Detection performance is evaluated using accuracy, sensitivity, specificity, and area under the receiver operating characteristic curve (AUC). These measures characterize overall classification correctness, the ability to identify attacks, resistance to false alarms, and the tradeoff between true-positive and false-positive behavior. The numerical performance obtained from these experiments is intentionally reserved for Section 4.1.2, where the results of the adaptive DBSCAN contribution are presented and compared with the earlier fixed-threshold framework.

3.1.3 Contribution 3: Time-Series-Based Detection of Multiple Small Biased GNSS Spoofing Attacks

The third research contribution extends the displacement-based spoofing detection framework by explicitly modeling the temporal evolution of displacement errors. This contribution was published in the IEEE Open Journal of Vehicular Technology as “Detection of Multiple Small Biased GPS Spoofing Attacks on Autonomous Vehicles Using Time Series Analysis” [12]. Whereas the preceding contributions detect anomalies from instantaneous residual magnitude or adaptive density structure, the time-series framework analyzes a recent history of displacement residuals and estimates the probability that an ongoing pattern is associated with GNSS spoofing. The principal objective is to detect multiple small biased attacks whose individual manipulation steps remain below conventional detection thresholds but whose cumulative effect can substantially distort the perceived vehicle trajectory.

The methodology uses a two-stage architecture. In the first stage, the previously developed data-driven displacement predictor estimates nominal short-horizon vehicle motion from trusted onboard measurements and forms a displacement residual relative to GNSS-reported motion. In the second stage, a dedicated attack probability predictor analyzes a fixed-length temporal window of the residual sequence. Two temporal configurations are investigated: 250 processed samples, corresponding to 25 seconds of recent history, and 500 processed samples, corresponding to 50 seconds. The longer window is used to examine smaller and more slowly accumulating biases, while the shorter window provides a more responsive representation for larger per-step biases. A latency-aware target-generation procedure and a fixed 50% probability decision threshold are incorporated to support real-time binary attack decisions.

3.1.3.1 Motivation for Temporal Attack Detection

The displacement-based and adaptive DBSCAN frameworks provide effective mechanisms for detecting abrupt or structured spoofing behavior; however, neither approach explicitly represents the temporal development of an attack. This limitation is important for multiple small biased spoofing, because a sophisticated attacker can introduce a sequence of small position manipulations that remain individually consistent with the nominal sensor uncertainty. When each residual is evaluated independently, the early steps of such a sequence may appear normal even though the accumulated position shift becomes operationally significant over time.

Temporal analysis addresses this limitation by treating the residual history as a structured signal rather than a set of independent measurements. The recent evolution of the displacement error

provides information about persistence, direction of change, and gradual accumulation that is not available from a single time step. The time-series approach therefore creates a short-term memory of vehicle behavior and enables the detector to learn patterns produced by sustained spoofing while remaining tolerant of isolated nominal fluctuations. This design is particularly relevant to the threat model considered in this project, where the attacker intentionally keeps individual spoofing steps below the pre-extracted displacement threshold.

A second motivation is the requirement for clear real-time decisions in a safety-critical navigation system. The proposed framework produces a direct spoofing probability and applies a hard 50% threshold to the output. The probability is not subjected to delayed confirmation, temporal smoothing, or a secondary confidence margin. The intent is to produce a deterministic attack/no-attack decision as soon as the learned temporal signature becomes sufficiently strong. The resulting architecture separates motion estimation from temporal attack inference and permits each stage to be understood and evaluated independently.

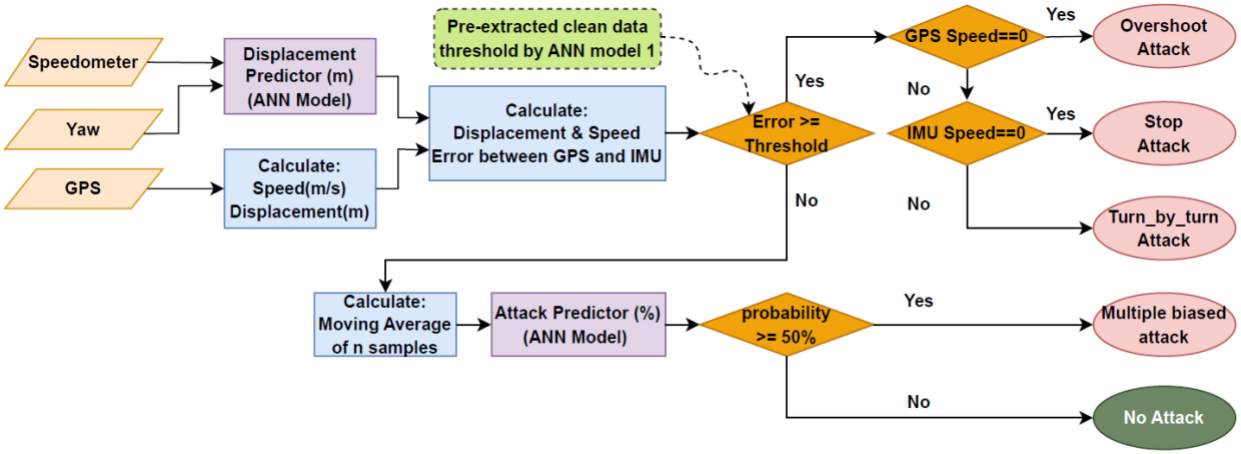


Figure 3.6. Two-stage GNSS spoofing detection framework based on time-series analysis. The displacement predictor model generates displacement residuals from GNSS and onboard sensor data, which are then processed by the attack probability predictor model to produce a real-time binary spoofing decision using a strict 50% threshold.

As illustrated in Figure 3.6, the conventional displacement and speed threshold branch remains available for turn-by-turn, stop, and overshoot attacks that produce sufficiently large inconsistencies. When the displacement error remains below the conventional threshold, the temporal branch analyzes the residual history and predicts the probability of a multiple small biased attack. This arrangement preserves the established attack-classification logic while adding a dedicated mechanism for below-threshold, gradually accumulating manipulations.

3.1.3.2 Data-Driven Displacement Predictor

The displacement predictor used in this contribution is the same data-driven motion model introduced in Contribution 1. No new sensing modality is introduced and the time-series contribution does not depend on a modified vehicle-motion model. Vehicle speed, heading information derived from the gyroscope, and the corresponding time intervals are used to estimate short-horizon planar displacement independently of GNSS. The GNSS measurements are processed in parallel to obtain the displacement perceived by the navigation receiver.

The purpose of reusing the existing displacement predictor is methodological isolation. The model provides the nominal motion estimate from which the scalar displacement residual is calculated, while the new contribution is concentrated in the temporal analysis of that residual. Accordingly, the predictor architecture, feature construction, and short-horizon displacement formulation documented in Section 3.1.1 are not repeated here. The residual defined in Eqs. (3.25) and (3.26) serves as the principal signal supplied to the time-series detection stage.

For conventional attacks whose displacement or speed errors exceed the clean-data thresholds, the rule-based detection and attack-identification procedure described previously remains applicable. For below-threshold events, the residual is not immediately interpreted as normal. Instead, successive residual samples are retained as a temporal sequence and passed to the attack probability predictor. This distinction is central to Contribution 3 because it allows the same motion-estimation foundation to support both instantaneous and temporal anomaly detection.

3.1.3.3 Displacement Error Time Series

The time-series input is constructed from the displacement error between the data-driven motion estimate and the corresponding GNSS-reported displacement. During model-development data preparation, the same underlying driving segment is processed in two forms. The first form is maintained as clean data, while the second form is generated by contaminating the GNSS measurements with multiple small biased spoofing attacks. The onboard sensor measurements are kept unchanged so that differences between the two residual sequences are attributable to the injected GNSS manipulation rather than to changes in the physical vehicle motion.

For the clean sequence, the displacement residual is calculated at each processed sample and an n-sample moving average is applied. This operation produces a recent-memory representation of nominal error behavior. The identical procedure is applied to the spoofed sequence, producing a second temporally averaged residual signal. Because individual spoofing steps are intentionally small, their effect may be difficult to distinguish from nominal uncertainty in a single residual sample. Repeated biases, however, alter the temporal evolution of the averaged residual and create a persistent divergence from the clean behavior.

The difference between the averaged clean and spoofed residual sequences is referred to as the differential error. This differential signal emphasizes the cumulative effect of the repeated spoofing biases and is used during supervised training to construct the target feature for the attack probability predictor. The parallel clean/spoofed processing is therefore a model-development procedure for generating labeled training information. During online operation, the trained attack probability predictor receives the recent history of the incoming displacement-error signal and does not require a simultaneously available clean duplicate of the attacked trajectory.

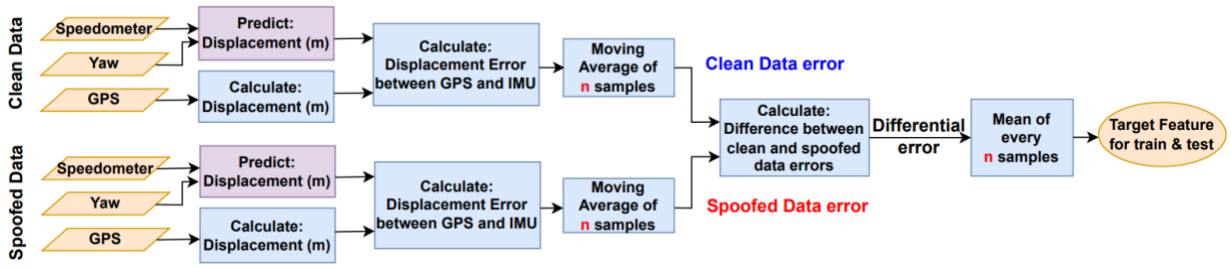


Figure 3.7. Feature construction process for training and testing the attack probability predictor model. Clean and spoofed displacement residuals are averaged using an n-sample window, their difference forms a differential error signal, and the block-wise mean of the differential error is used to construct the supervised target feature.

Figure 3.7 summarizes the complete feature-generation sequence. This construction moves the detection problem from an instantaneous error comparison to a temporal pattern-recognition problem. It also provides a controlled mechanism for associating the observed residual history with the presence or absence of spoofing during supervised model development.

3.1.3.4 Time-Window Construction

The temporal memory of the detector is determined by the number of processed residual samples included in each window. The sensor streams originate from HDD at 100 Hz and are processed using the same one-in-ten sampling strategy described in Section 3.1.1, resulting in an effective 10-Hz sequence. Consequently, a 250-sample window represents approximately 25 seconds of recent behavior, whereas a 500-sample window represents approximately 50 seconds.

Two separate configurations are used because the temporal characteristics of a spoofing attack depend on the magnitude of its individual bias steps. A shorter 250-sample window provides less temporal memory but greater responsiveness to a sequence whose per-step displacement change is comparatively larger. The 500-sample window provides twice the temporal context and is intended to capture smaller biases that accumulate more slowly and can remain concealed within nominal displacement uncertainty for a longer period.

During feature generation, the n-sample moving average creates a temporally smoothed representation of the displacement error. The differential error obtained from the clean and spoofed averaged sequences is then divided into n-sample intervals, and the mean associated with each interval is used to construct the supervised target feature. The same value of n is therefore used consistently to define the recent-memory window, the moving-average operation, and the latency-aware target-generation interval. Separate model-development pipelines are maintained for n = 250 and $n = 500$ rather than combining the two temporal scales into a single predictor.

This explicit encoding of recent history allows the temporal dependence to be represented before the data reach the neural network. As a result, a recurrent architecture is not required to maintain an internal memory state. The network can operate as a feedforward predictor on a fixed-length feature vector while still exploiting information accumulated across the selected time horizon.

3.1.3.5 Probabilistic Attack Prediction

The second stage of the framework is implemented using a feedforward Artificial Neural Network (ANN) that maps the recent displacement-error history to a scalar attack probability. A feedforward architecture was selected instead of a recurrent neural network because the temporal dependency is explicitly encoded in the fixed-length input window. This avoids sequential unrolling and internal state maintenance, reducing memory requirements and inference complexity for embedded automotive implementation.

Separate ANN models are developed for the $n = 250$ and $n = 500$ configurations. For $n = 250$, the first and second hidden layers contain 250 neurons each, followed by layers containing 125 and 10 neurons. For n = 500, the first and second hidden layers contain 500 neurons each, followed by the same 125-neuron and 10-neuron layers. Each model uses a single-neuron output layer. Rectified Linear Unit (ReLU) activations are applied to the hidden layers, while a sigmoid activation is applied to the output so that the predicted attack probability remains in the interval from 0 to 1. The architectural and training parameters are summarized in Table 3.2.

Table 3.2. Neural network architecture and training hyperparameters for the attack probability predictor models used in the multiple small biased attack scenarios.

Hyperparameter	n = 250	n = 500
Neurons in 1st hidden layer	250	500
Neurons in 2nd hidden layer	250	500
Neurons in 3rd hidden layer	125	125
Neurons in 4th hidden layer	10	10
Output layer neurons	1	1
Activation (hidden layers)	ReLU	ReLU
Activation (output layer)	Sigmoid	Sigmoid
Optimizer	Adam	Adam
Batch size	32	32
Training epochs	100	100

The attack probability predictor is trained in a supervised manner using clean and spoofed time-series sequences constructed through the procedure described above. In the published time-series study, the attack-predictor dataset contains 120,000 samples and is divided using an 80/20 training and testing split, resulting in 96,000 training samples and 24,000 testing samples [12]. The Adam optimizer and a batch size of 32 are used for 100 epochs. Mean Absolute Error (MAE) is used as the training loss and Mean Squared Error (MSE) is monitored as an additional evaluation metric during model development.

The single output neuron represents the predicted probability of an ongoing spoofing attack rather than a displacement quantity. This distinction is important: the first neural model estimates nominal vehicle motion, whereas the second neural model performs temporal attack inference. The two models therefore solve different tasks and are trained using different target variables even though they are connected sequentially within the overall detection framework.

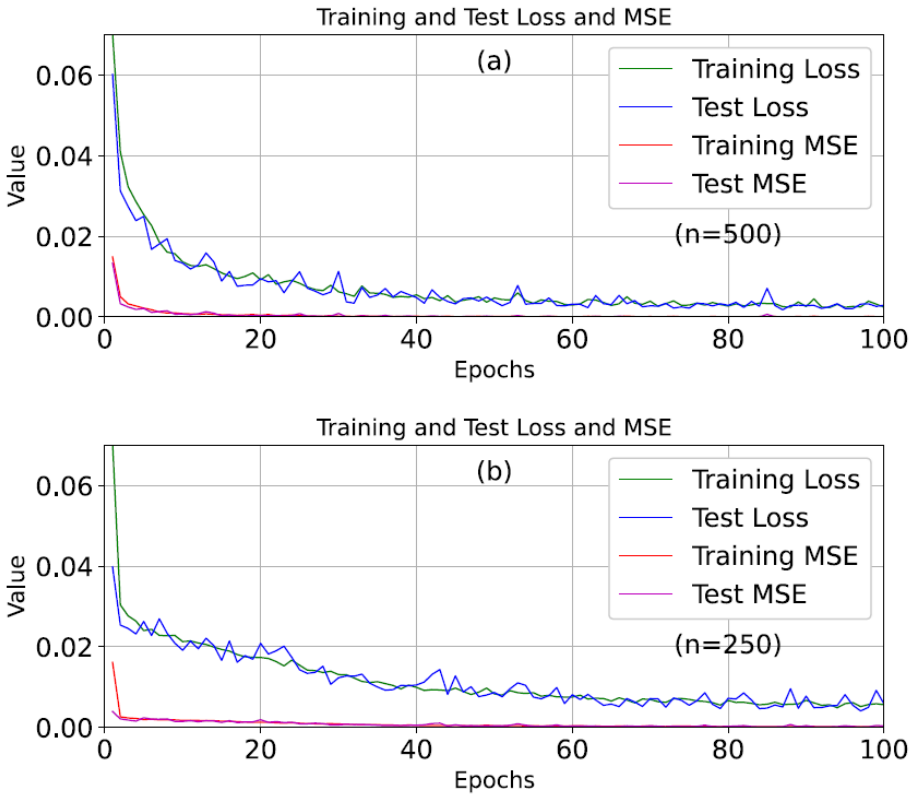


Figure 3.8. Training process of the attack probability predictor model for multiple small biased attacks. The figure illustrates the convergence behavior of the training and validation loss for the $n = 250 and n = 500$ configurations.

The training curves are used to verify convergence of both temporal configurations. Quantitative spoofing-detection performance is not reported in this methodology section and is reserved for Section 4.1.3 so that model-development procedures remain separated from the subsequent results analysis.

3.1.3.6 Detection at Different Statistical Attack Magnitudes

The two temporal configurations are evaluated using attack steps defined relative to the nominal standard deviation of the displacement residual. In the time-series study, the standard deviation of the clean GPS-versus-onboard displacement error is approximately 0.187 m [12]. This statistical characterization is used to construct two increasingly stealthy multiple-small-biased attack scenarios while keeping each individual manipulation well below the conventional displacement threshold of 1.79 m.

For the n = 250 configuration, the GNSS position is manipulated through 250 consecutive small biased steps. Each step has a Euclidean displacement magnitude of approximately 0.56 m, which corresponds to approximately three times the nominal residual standard deviation. Repetition of these small steps creates an accumulated localization shift of approximately 141 m over 25 seconds. Although the cumulative displacement is large, the magnitude of each individual step remains substantially below the conventional 1.79-m threshold.

The $n = 500$ configuration represents a more gradual and stealthier attack. The same approximate 141-m accumulated localization shift is distributed across 500 consecutive manipulations, reducing the per-step displacement magnitude to approximately 0.28 m. This value corresponds to approximately 1.5 times the nominal residual standard deviation. Because the individual steps are considerably smaller and are introduced over a 50-second interval, the attack remains close to the natural error distribution for a longer period and requires a longer temporal context for reliable identification.

The two configurations therefore evaluate complementary operating points of the detector. The 25-second case examines whether temporal analysis can identify a sequence of moderately small but persistent errors, while the 50-second case examines sensitivity to smaller and more slowly accumulating deviations. In both cases, the attack design separates instantaneous detectability from cumulative navigation consequence and provides a controlled test of the benefit gained from time-series information.

3.1.3.7 Latency Compensation and Real-Time Detection

A moving-average operation necessarily introduces temporal lag between the true onset of an attack and the corresponding change in the averaged displacement-error signal. If the delayed signal were used directly as the supervised target, the attack probability predictor could learn a shifted representation of the attack and consequently generate late detections. The time-series framework therefore incorporates a latency-aware target-generation strategy based on the mean of the differential-error signal over the same n-sample interval used to construct the temporal features.

The block-wise mean of the differential error is used to align the supervised target more closely with the actual temporal evolution of the spoofing sequence. In the source study, this operation is described as a moving-mean compensation for the latency created by the n-sample moving average. The resulting target feature is used during training and testing of the attack probability predictor so that the learned probability transition is synchronized with the attack interval rather than with the delayed moving-average response.

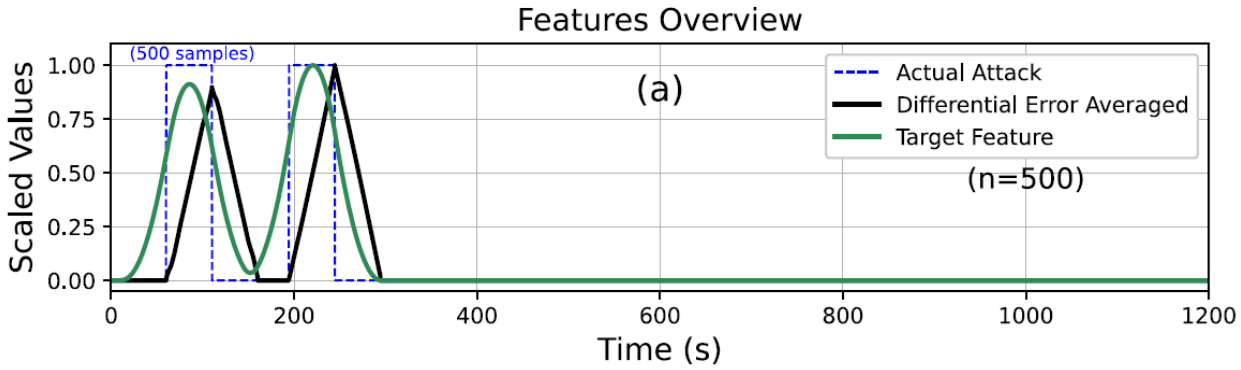


Figure 3.9. Target feature used for supervised training of the attack probability predictor model. The block-wise mean of the differential error is used to compensate for the temporal lag introduced by the recent-memory moving-average operation and to construct the target feature for probability prediction.

During online detection, the trained ANN produces one probability value in the range from 0 to 1 for each evaluated temporal window. A fixed decision threshold of 50% is then applied. A predicted probability greater than 50% is classified immediately as an attack; otherwise the current condition is classified as no attack. No additional smoothing, hysteresis, confidence margin, or delayed confirmation is applied after the probability prediction.

The real-time implementation is supported by two design choices. First, temporal memory is represented explicitly through fixed-length feature construction rather than through recurrent hidden states. Second, the probability output is evaluated directly using a fixed decision rule. These choices keep the inference stage lightweight and deterministic. The temporal window itself still represents the recent history required to identify a gradual attack, but once the required recent-memory feature is available, the decision is generated without an additional post-processing delay. The real-time probability trajectories and detection outcomes are presented later in Section 4.1.3.

3.1.3.8 Evaluation Dataset and Experimental Design

The time-series framework is evaluated using the Honda Research Institute Driving Dataset (HDD) [14]. The same general sensor sources and preprocessing principles used in the preceding contributions are retained so that the evaluation remains grounded in naturalistic vehicle operation. HDD contains synchronized GNSS, inertial, and CAN-bus measurements collected under urban, suburban, and highway driving conditions. The raw sensor streams are recorded at 100 Hz, and the detection pipeline uses the previously established processed sampling sequence to construct the temporal residual windows.

Five driving datasets are selected to provide variation in driving environment, vehicle speed, time of day, and motion behavior. Before attack generation, synchronization, continuity, data integrity, timestamp alignment, data representation, and anomalous or missing records are checked as part of the preprocessing procedure. Spoofing scenarios are then generated by modifying the GNSS measurements while retaining the corresponding onboard vehicle measurements as the trusted physical-motion reference.

The evaluation includes turn-by-turn, stop, overshoot, and multiple small biased spoofing. For each conventional attack type, five attack realizations are generated in each of the five selected datasets with randomized attack timing, duration, and, where applicable, magnitude, producing 25 trials per attack category. Turn-by-turn and overshoot attacks are mounted during vehicle motion, whereas stop attacks are mounted in stationary segments. The GNSS trajectories are manipulated so that the resulting motion remains sufficiently smooth to avoid creating trivial discontinuities.

Multiple small biased attacks are evaluated separately for the $n = 250$ and $n = 500$ configurations. Each configuration is mounted five times in each of the five driving datasets, producing 25 independent realizations per temporal configuration. The $n = 250$ case uses approximately 0.56-m attack steps over 250 processed samples, while the $n = 500$ case uses approximately 0.28-m steps over 500 processed samples. Both scenarios are constructed to produce an accumulated displacement of approximately 141 m while maintaining each individual manipulation below the conventional system threshold.

The attack probability predictor models are trained and tested on labeled time-series sequences using the 80/20 development split reported in the published study. Performance is assessed using

accuracy, F1-score, sensitivity, specificity, and area under the receiver operating characteristic curve (AUC). These metrics quantify overall classification correctness, balance between precision and recall, attack-detection sensitivity, resistance to false alarms, and discrimination capability. All decisions are interpreted using the fixed 50% probability threshold so that the reported performance reflects the same binary rule intended for online operation.

The numerical detection results for the two time-series configurations and the conventional spoofing scenarios are intentionally deferred to Section 4.1.3. This separation allows the present section to document the temporal feature construction, model architecture, latency compensation, attack mounting strategy, and evaluation protocol without mixing methodological definitions with performance outcomes.

3.1.4 Contribution 4: Stereo Visual Navigation in GNSS-Compromised and Denied Environments

This fourth contribution extends the Detection and Identification subtask from recognizing compromised GNSS information toward maintaining a navigation capability after GNSS can no longer be treated as a trusted source. The work is represented by the current stereo visual navigation (VN) manuscript and is ongoing research at the time of this final report. It has not yet been published or peer reviewed. Accordingly, this section documents the current methodology and experimental configuration, while the latest quantitative results are reserved for Section 4.1.4 and should be updated if the manuscript changes before final submission.

The current VN framework estimates short-horizon planar displacement directly from synchronized stereo images. It does not attempt to replace GNSS with a complete mapping or full-state estimation system. Instead, visual correspondences observed over a short temporal horizon are converted into a compact geometric representation, the directions of the global displacement components are classified, and specialized Extreme Gradient Boosting (XGBoost) models are used to predict displacement magnitude. This formulation is intended to provide a modular bridge between cyberattack detection and GNSS-independent motion estimation.

The methodology is visual-only in its present form. Although an IMU is installed on the experimental vehicle and inertial measurements are recorded during data collection, those measurements are not used as inputs to the VN displacement predictor described in this contribution. The distinction is important because visual-inertial navigation is treated as future work rather than as a demonstrated component of the present CARMEN+ contribution.

3.1.4.1 Motivation for Post-Detection Navigation

The preceding contributions focus on determining when GNSS information has become inconsistent with trusted onboard sensing. Once an attack is detected, however, the navigation problem changes: the compromised GNSS measurements should not continue to be used as the primary localization source. A resilient automated transportation system therefore benefits from a complementary onboard method that can estimate vehicle motion without relying on the GNSS signal that triggered the cybersecurity alarm.

Camera-based navigation is attractive for this role because visual measurements are generated locally and contain information about the relative motion of the vehicle with respect to the

surrounding environment. Visual odometry (VO) and visual simultaneous localization and mapping (SLAM) have been widely studied for vehicle navigation, and stereo configurations are particularly useful because disparity provides a metric depth cue without the scale ambiguity associated with a single monocular camera [16], [17]. However, many full VO and SLAM pipelines require iterative optimization, keyframe management, dense image processing, map maintenance, or deep sequential models. The ongoing VN work therefore investigates a narrower short-horizon displacement objective that can be represented using a small number of sparse stereo correspondences.

The current framework predicts the planar global displacement between synchronized stereo observations acquired at time steps k and k+n. The temporal horizon is defined as $n \in \{1, 2, 3, 4, 5\}$. Because stereo images are recorded at 20 Hz, these horizons correspond to frame separations from 50 ms to 250 ms. The prediction target is local in time: each sample describes the displacement between two observations rather than an absolute global position. This reduces dependence on long temporal histories and avoids requiring a recurrent neural-network state or a full trajectory optimization process for each prediction.

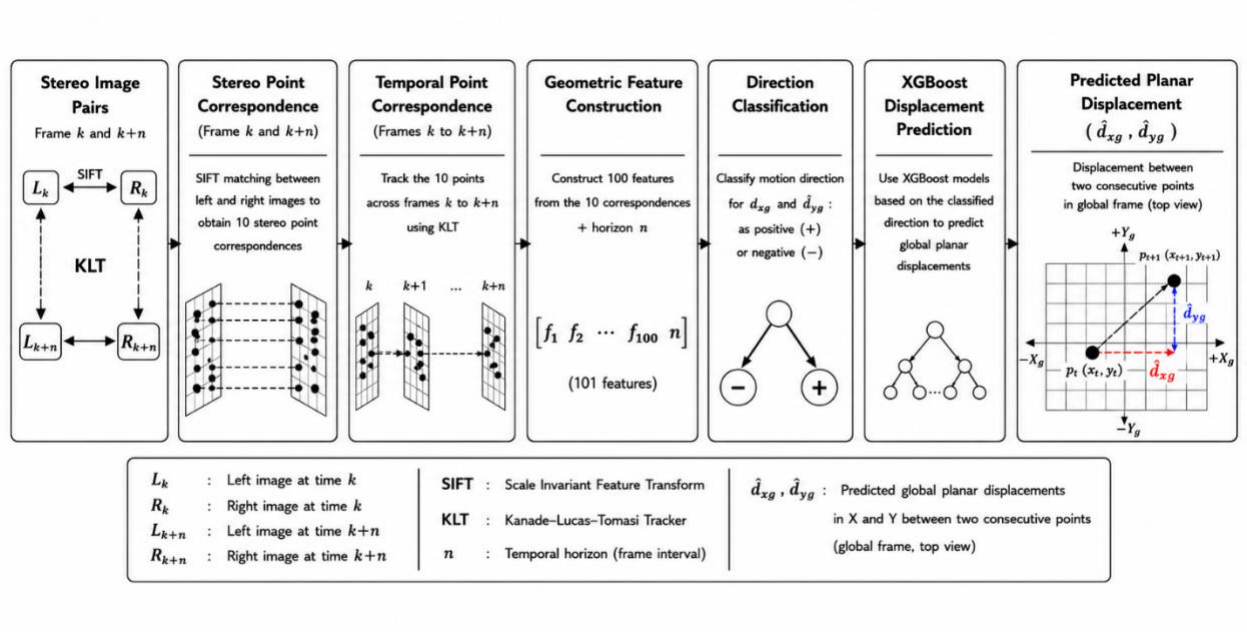


Figure 3.10. Overall framework of the ongoing stereo visual navigation method for short-horizon planar global displacement prediction. Stereo correspondence, chained temporal tracking, compact geometric feature construction, direction classification, and conditional XGBoost prediction are applied sequentially.

As summarized in Figure 3.10, the processing chain is intentionally modular. Stereo point correspondence and temporal tracking are separated from geometric feature construction and from the machine-learning stages. This organization makes it possible to evaluate the quality and computational requirements of each stage independently and provides a practical structure for future deployment on edge computing hardware.

3.1.4.2 Experimental Vehicle and Stereo Camera Platform

The ongoing VN framework is developed and evaluated using an indoor experimental testbed consisting of a miniature urban environment, an instrumented differential-drive robotic vehicle, a VICON motion-capture system, and a synchronized data-acquisition computer. The miniature city contains roads, intersections, buildings, curved segments, and path-crossing sections that create repeated changes in vehicle motion and camera viewpoint. These layouts provide feature-rich scenes while maintaining controlled ground-truth measurement conditions.

The experimental vehicle has an overall length of approximately 225 mm and contains separate data-acquisition and driving-control subsystems. A Raspberry Pi 4 is used as the onboard data-acquisition computer. Synchronized left and right camera streams are received through USB and recorded for subsequent processing. The driving subsystem uses three line-following sensors positioned under the vehicle, an Arduino Uno controller, a motor-driver module, and four drive motors. This arrangement enables repeatable automated traversal of the road layouts during data collection.

The stereo sensor is a binocular USB camera module based on the OmniVision OV9281 monochrome global-shutter sensor with dual fixed-focus M12 lenses. The two cameras are rigidly mounted with a 60 mm baseline. Although the camera hardware supports frame rates up to 120 Hz, synchronized stereo images are recorded at 20 Hz for the experiments. The global-shutter configuration is useful for a moving platform because both image sensors capture the scene without the line-by-line temporal distortion associated with a rolling shutter.

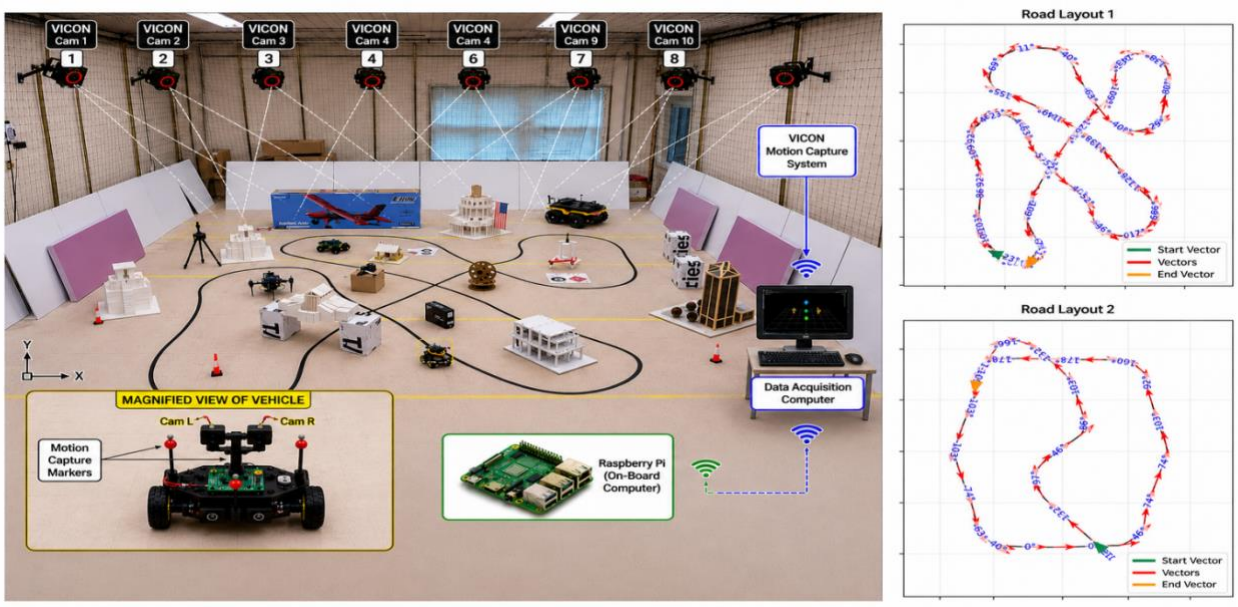


Figure 3.11. Experimental platform for the ongoing VN study. The miniature urban environment is instrumented with VICON motion capture and includes the stereo-equipped vehicle and representative road layouts used to generate diverse motion trajectories.

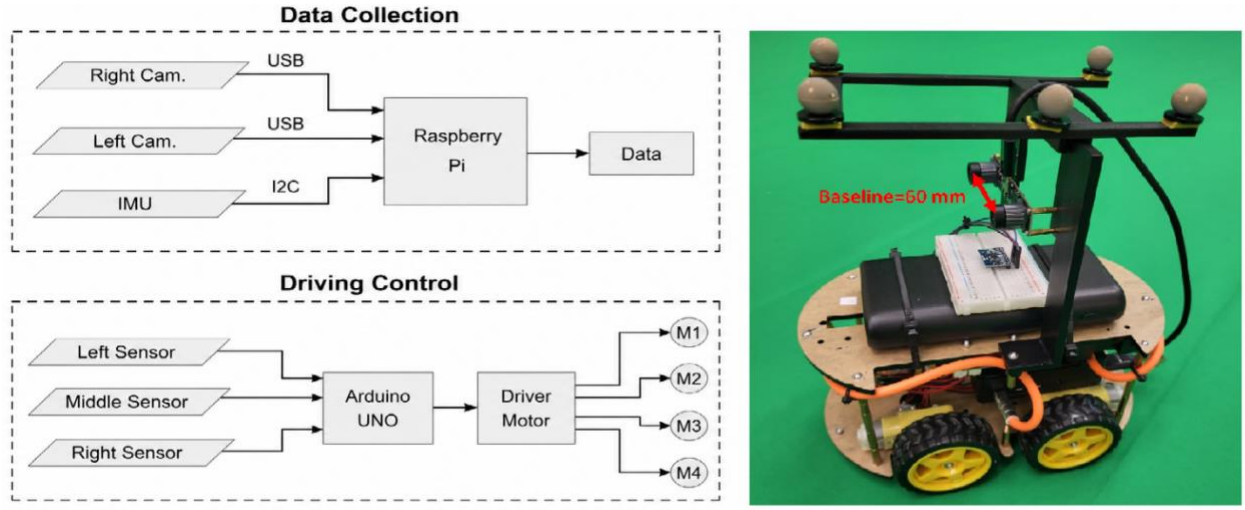


Figure 3.12. Vehicle platform architecture and hardware implementation for stereo data acquisition, automated driving control, and VICON tracking.

Multiple reflective markers are mounted on the upper frame so that the vehicle can be continuously observed by the VICON system. Stereo images, inertial measurements, and VICON measurements are recorded during the same experiments. For the present VN methodology, the stereo imagery provides the prediction input and VICON provides the supervised displacement target; the recorded inertial measurements are retained for subsequent visual-inertial research.

3.1.4.3 VICON Ground-Truth System

Vehicle position and orientation ground truth are obtained from the VICON motion-capture system. Reflective markers attached to the vehicle are tracked by cameras distributed around the experimental field. VICON measurements are acquired at 100 Hz, providing a dense and smooth record of the vehicle motion for supervised model development and evaluation.

Stereo images are acquired at 20 Hz and are synchronized with the 100-Hz VICON stream using timestamp-based alignment. Each selected stereo image observation is associated with the corresponding VICON measurement in the unified dataset. For an image pair separated by temporal horizon n, the global displacement targets are computed from the corresponding VICON positions at the beginning and end of the interval as

$$\begin{aligned} dx_g &= x_{k+n} - x_k, \\ dy_g &= y_{k+n} - y_k. \end{aligned} \tag{3.40}$$

where the two terms are the ground-truth global x-axis and y-axis displacement components expressed in the VICON reference frame. Each learning sample therefore pairs the visual feature vector constructed from the stereo images with the corresponding VICON displacement target. VICON is required only during controlled data collection and model evaluation; it is not an input to the visual displacement predictor during inference.

3.1.4.4 Stereo SIFT Feature Correspondence

The visual processing stage begins with sparse stereo correspondence at time step k. Let L_k and R_k denote the synchronized left and right images. Distinctive image keypoints and descriptors are

generated using the Scale-Invariant Feature Transform (SIFT) [18]. Candidate left-right matches are obtained from descriptor similarity and are subsequently filtered to reject correspondences that are inconsistent with the expected stereo geometry.

The current implementation uses a Lowe descriptor-ratio threshold of 0.85. Image regions near the boundaries are excluded using a region of interest (ROI) trim fraction of 0.08, and only disparities between 1 and 200 pixels are retained. Epipolar consistency is evaluated by estimating a fundamental matrix using Random Sample Consensus (RANSAC) [22]; correspondences with an average epipolar distance greater than 2 pixels are rejected. Duplicate points are removed with a 0.5-pixel tolerance.

For a valid stereo correspondence, disparity is defined from the horizontal pixel coordinates of the left and right observations as

$$\text{disparity}_{t,i} = u_{L,t,i} - u_{R,t,i}, \qquad t \in \{k, k+n\} \tag{3.41}$$

where the left and right terms denote the horizontal image coordinates of the i-th matched stereo feature at time t. Disparity is not converted into a full dense depth image in the proposed learning framework. Instead, disparity and inverse disparity are retained as compact geometric cues because stereo depth is inversely related to disparity. This avoids the computational cost of dense stereo reconstruction while preserving information related to feature distance from the cameras.

After stereo filtering, the remaining matches are spatially distributed across the image. A minimum spacing of 50 pixels is initially enforced between selected left-image points. If this criterion yields insufficient candidates, the spacing is relaxed to 30 pixels. The spatial-selection step reduces concentration of features in a small image region and improves the geometric diversity of the point set supplied to temporal tracking.

3.1.4.5 Chained KLT Temporal Tracking

Temporal correspondence is established using the Kanade-Lucas-Tomasi (KLT) tracker [19], [20]. Rather than tracking a point directly from frame k to a potentially more distant frame k+n, the proposed chained formulation advances the selected left-image points sequentially through each intermediate frame. This procedure preserves the local-motion assumption of KLT and improves correspondence continuity over the five temporal horizons considered in the study.

The chained KLT implementation uses a 40 × 40 pixel base tracking window. A forward-backward consistency test is applied with a threshold proportional to the temporal horizon, defined in the current configuration as $30\sqrt{n}$ pixels. Candidate count is expanded using a multiplier of 20 before temporal tracking so that sufficient points remain after all stereo and temporal consistency checks.

At time step k+n, SIFT-based stereo correspondence is performed again between the synchronized left and right images. Each successfully tracked left-image point is associated with the nearest filtered stereo correspondence at the same time step, subject to a 25-pixel association threshold and a uniqueness requirement. A complete stereo-temporal correspondence for feature i is therefore represented by the quadruple

$$Q_i = \left(L_k^i, R_k^i, L_{k+n}^i, R_{k+n}^i\right) \tag{3.42}$$

where the four observations represent the left and right stereo positions at the beginning and end of the temporal horizon. The first ten quadruples satisfying the stereo consistency, spatial diversity, chained temporal tracking, and final stereo association constraints are retained:

$$\mathcal{Q} = \{Q_1, Q_2, \dots, Q_{10}\} \tag{3.43}$$

Restricting the representation to ten valid quadruples provides a fixed-size input for the subsequent machine-learning stages. The current correspondence and tracking parameters are summarized in Table 3.3. These values document the configuration used in the ongoing manuscript and can be refined in future experiments if the camera geometry or operating environment changes.

Table 3.3. Stereo correspondence, temporal tracking, and quadruple-selection parameters used in the current VN configuration.

Parameter	Value
Lowe ratio threshold	0.85
ROI trim fraction	0.08
Minimum disparity	1 pixel
Maximum disparity	200 pixels
Epipolar distance threshold	2 pixels
Duplicate tolerance	0.5 pixels
Minimum spatial spacing	50 pixels
Relaxed spatial spacing	30 pixels
Candidate multiplier	20
Base KLT window size	40 × 40 pixels
KLT forward-backward threshold	$30\sqrt{n}$ pixels
Stereo association threshold at k+n	25 pixels
Number of retained quadruples	10

3.1.4.6 Geometric Feature Construction

Each retained stereo-temporal quadruple is converted into ten engineered geometric features. The objective is to represent three complementary sources of visual motion information in a compact tabular form: image location, stereo depth cue, and temporal image displacement. Because the resulting representation is fixed in size, it can be processed directly by tree-based machine-learning models without storing raw image patches or long image sequences.

For the i-th quadruple, temporal image displacement is computed from the left-image coordinates at the beginning and end of the horizon as

$$\begin{aligned} du_i &= u_{k+n,i} - u_{k,i}, \\ dv_i &= v_{k+n,i} - v_{k,i}. \end{aligned} \tag{3.44}$$

The two displacement terms capture optical motion in the image plane. To incorporate a depth-related cue, inverse disparity is calculated at both time steps. Two additional interaction terms

multiply the temporal image displacement components by the inverse disparity at time k. The complete ten-element feature vector for the i-th quadruple is

$$\mathbf{f}_i = \left[u_{k,i}, v_{k,i}, u_{k+n,i}, v_{k+n,i}, \frac{1}{\text{disparity}_{k,i}}, \frac{1}{\text{disparity}_{k+n,i}}, du_i, dv_i, \frac{du_i}{\text{disparity}_{k,i}}, \frac{dv_i}{\text{disparity}_{k,i}}\right] \quad (3.45)$$

The interaction terms couple apparent image motion with stereo geometry. A feature moving by the same number of pixels can correspond to different physical vehicle motion depending on its depth; weighting temporal motion by inverse disparity provides the learning model with information related to this dependence without requiring a full three-dimensional reconstruction.

Ten quadruples contribute 100 point-based features. The temporal horizon n is appended as the final feature so that one set of classifiers and predictors can distinguish observations collected over different frame separations. The final input vector is therefore

$$\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_{10}, n] \in \mathbb{R}^{101} \quad (3.46)$$

The 101-dimensional representation is summarized in Table 3.4. The same feature vector is used by both the direction classifiers and the conditional displacement predictors, maintaining a common visual representation across the learning pipeline.

Table 3.4. Summary of the geometric features constructed from ten stereo-temporal quadruples.

Feature group	Number of features
Image coordinates at k and $k+n$	40
Inverse-disparity terms	20
Temporal image-displacement terms	20
Motion-depth interaction terms	20
Temporal horizon n	1
Total	101

3.1.4.7 Global Displacement Direction Classification

The current VN framework separates displacement direction from displacement magnitude. Before regression is performed, the signs of dx_g and dy_g are classified independently. This decomposition is motivated by the observation that a single regression model trained across both positive and negative motion must represent multiple directional regimes simultaneously, whereas direction-conditioned models can learn from more homogeneous subsets of the visual feature space.

A ±5 mm transition region is defined around zero displacement for classifier development. Samples inside this interval represent minimal physical motion relative to the approximately 225 mm vehicle length and are also more sensitive to image localization, stereo correspondence, disparity, and synchronization uncertainty. Direction classification is therefore trained using samples outside this transition region. The binary labels are defined as

$$C_{dx} = \begin{cases} 1, & dx_g \geq 0 \\ 0, & dx_g < 0 \end{cases}, \qquad C_{dy} = \begin{cases} 1, & dy_g \geq 0 \\ 0, & dy_g < 0 \end{cases} \tag{3.47}$$

where a class value of 1 represents positive global displacement and a class value of 0 represents negative global displacement. The same 101 visual features are supplied to two independent XGBoost classifiers [21], one for dx_g and one for dy_g. The classifier datasets are partitioned using a stratified 70% training, 15% validation, and 15% testing split so that class balance is maintained across the three subsets.

During model development, binary logloss and classification error are monitored and separate hyperparameter configurations are used for the two axes. Table 3.5 records the current classifier configuration from the ongoing manuscript. The held-out test data are not used for parameter selection; their quantitative performance is reserved for Section 4.1.4.

Table 3.5. XGBoost hyperparameters for global displacement direction classification in the current VN configuration.

Parameter	dx_g**classifier**	dy_g**classifier**
Objective	binary: logistic	binary: logistic
Number of trees	1400	1200
Maximum depth	6	6
Learning rate	0.018	0.014
Minimum child weight	5	6
Gamma	0.12	0.15
Subsample ratio	0.90	0.84
Column sampling ratio	0.90	0.84
L2 regularization (λ)	5.5	8.0
L1 regularization (α)	0.2	0.4
Classification threshold	0.4850	0.5075
Evaluation metric	Binary logloss and error	Binary logloss and error

3.1.4.8 Conditional XGBoost Displacement Prediction

After direction classification, the displacement magnitude is predicted using four specialized XGBoost regression models [21]. Two models represent negative and positive dx_g motion and two models represent negative and positive dy_g motion. All four regressors use the same 101-dimensional feature vector. The only difference is the subset of training samples used to develop each model and the corresponding tuned hyperparameters.

The negative and positive predictors for each global displacement axis are trained separately. The output of each sign classifier selects which direction-conditioned regression model is evaluated:

$$\widehat{dx_g} = M_{dx}^{C_{dx}}(\mathbf{F}), \qquad \widehat{dy_g} = M_{dy}^{C_{dy}}(\mathbf{F}) \tag{3.48}$$

This conditional structure transforms one difficult signed regression problem into four narrower regression problems. Each predictor learns only one displacement direction, while the classifier

provides the routing decision. For each directional subset, the data are divided into 70% training, 15% validation, and 15% testing partitions. Training and validation data are used for parameter optimization and early stopping, while the test data remain unseen until final evaluation.

The current regression hyperparameters are listed in Table 3.6. The XGBoost squared-error objective is used for continuous displacement prediction, and root mean square error is monitored during training. Early stopping is applied after 160 rounds without validation improvement. These settings represent the present manuscript configuration and document the model-development procedure without introducing the final prediction results into the methodology chapter.

Table 3.6. XGBoost predictor hyperparameters for the four direction-conditioned global displacement models.

Parameter	**Neg.** dx_g	**Pos.** dx_g	**Neg.** dy_g	**Pos.** dy_g
Number of trees	1800	1600	1800	1800
Maximum depth	5	5	4	4
Learning rate	0.016	0.018	0.020	0.020
Subsample ratio	0.80	0.78	0.86	0.86
Column sampling ratio	0.75	0.72	0.84	0.84
Minimum child weight	8	8	6	6
Gamma	0.18	0.20	0.10	0.10
L2 regularization	10.0	12.0	7.0	7.0
L1 regularization	0.8	1.0	0.4	0.4
Objective	SE	SE	SE	SE
Evaluation metric	RMSE	RMSE	RMSE	RMSE
Early stopping rounds	160	160	160	160

3.1.4.9 Short-Horizon Global Displacement Estimation

The output of the conditional framework is a two-component planar displacement increment expressed in the global VICON coordinate frame. Once the classifier-selected regressors produce their axis-specific estimates, the predicted displacement vector is

$$\hat{\mathbf{d}}_g = \begin{bmatrix} \widehat{dx}_g \\ \widehat{dy}_g \end{bmatrix} \tag{3.49}$$

The prediction represents vehicle motion between two synchronized observations separated by horizon n rather than an absolute vehicle position. This definition is consistent with the broader displacement-based strategy used elsewhere in the CARMEN+ research: the learning task is locally defined over a finite interval, reducing the need to learn a long-duration absolute-position state.

For trajectory-level analysis, consecutive predicted displacement increments can be accumulated recursively from a known starting point according to

$$\hat{\mathbf{p}}(k+1) = \hat{\mathbf{p}}(k) + \hat{\mathbf{d}}_g(k) \tag{3.50}$$

where the first term denotes the reconstructed planar position and the second term is the predicted visual displacement increment. This recursive accumulation is used only for trajectory-level evaluation. The fundamental model output remains the short-horizon displacement increment, which can also be supplied to a higher-level localization or sensor-fusion architecture in future work.

3.1.4.10 Real-Time Implementation Considerations

The current VN framework is designed with real-time edge implementation in mind, but the ongoing manuscript does not yet report end-to-end execution time on a target embedded computing platform. The Raspberry Pi 4 used in the experiments functions as the image data-acquisition computer; the manuscript does not establish that the complete SIFT, KLT, feature-construction, classification, and XGBoost prediction pipeline has already been deployed and benchmarked on that device. For this reason, the present report treats real-time operation as an implementation objective rather than a demonstrated runtime result.

The 20-Hz stereo acquisition rate establishes a 50-ms interval between synchronized image pairs. For a strictly serial implementation that produces one new estimate for each incoming stereo pair, the complete online processing chain would need to sustain the corresponding throughput. The stages that must be benchmarked include SIFT key point detection and descriptor computation in both stereo pairs, descriptor matching and geometric filtering, chained KLT tracking through the required intermediate frames, final stereo association, construction of the 101 features, evaluation of the two direction classifiers, and evaluation of the two selected displacement regressors. Pipelining or parallel execution could alter the relationship between throughput and individual-sample latency, so both quantities should ultimately be measured on the intended hardware.

Several design choices reduce the computational scope relative to full dense or optimization-based visual navigation. Only ten stereo-temporal quadruples are retained for the learned representation, the predictor operates on a 101-element tabular feature vector, and XGBoost inference does not require recurrent state propagation or iterative nonlinear pose optimization. At the same time, sparse feature extraction and correspondence remain nontrivial image-processing operations, and their execution cost depends on image content, number of detected key points, temporal horizon, processor architecture, and software implementation. A defensible real-time claim therefore requires measured stage-level and end-to-end timing rather than relying only on algorithmic complexity.

Future implementation work should report average, median, and worst-case processing time for each online stage and for the complete pipeline, together with the target processor and level of parallelism. The present methodology provides the architecture needed for such benchmarking, while the current report avoids presenting unmeasured timing as demonstrated real-time performance. This distinction is especially important because the VN contribution remains ongoing and is expected to evolve before journal publication.

3.2 Subtask 2: Data-Driven Verification and Validation of Attacked HATS

Subtask 2 develops data-driven verification and validation (V&V) methods for assessing both the expected reliability of cyberattack-detection models and the timing of cyberattack effects after an

alarm has been raised. The first contribution evaluates the intrinsic separability of benign and malicious cyber-traffic data using fuzzy equivalence relations so that achievable anomaly-detection performance can be estimated before a trained model is accepted as reliable [25]. The second contribution addresses a complementary post-alarm problem: estimating when a deceptive GNSS attack actually began by analyzing residual sequences generated by a nonlinear navigation estimator [26].

The two contributions therefore operate at different stages of assurance. The fuzzy framework provides a dataset-structure-based benchmark for model validation, whereas the onset-localization framework reconstructs the beginning of a detected GNSS attack within the vulnerable interval preceding the alarm. The methodological details are presented below with the equations, datasets, model configurations, and evaluation procedures required to reproduce the reported studies.

3.2.1 Contribution 1: Fuzzy Equivalence Relation Framework for Anomaly Detection Accuracy Estimation

Contribution 1 is based on the fuzzy equivalence relation clustering framework presented at ARCI 2025 [25]. The framework is not designed as a new Distributed Denial of Service (DDoS) classifier. Instead, it analyzes the similarity structure of labeled benign and DDoS samples to estimate an accuracy range that can serve as an independent validation reference for data-driven anomaly-detection models. The method combines fuzzy tolerance relations, iterative max-min composition, alpha-cut defuzzification, and class-separation measures to quantify whether the available data support a reliable distinction between benign and malicious traffic [25], [28].

3.2.1.1 Problem Formulation and Validation Objective

Many anomaly-detection studies evaluate model reliability only after training by reporting accuracy, precision, recall, or related test-set measures [38]–[40]. Existing fuzzy approaches have used fuzzy inference, fuzzy C-means, kernel fuzzy clustering, and hybrid fuzzy-machine-learning designs directly for anomaly or DDoS detection [41]–[46]. The ARCI study identifies a complementary V&V question: before treating a trained detector as reliable, can the structure of the labeled dataset itself provide an independent estimate of how separable the benign and malicious classes are? The proposed framework addresses this question by measuring pairwise similarity among samples and progressively tightening the similarity requirement until a useful balance between within-class cohesion and between-class separation is obtained [25].

Let the dataset be represented by $X = x_1, x_2, \ldots, x_n$, where each sample x_i contains the retained continuous traffic features. The objective is to derive a relation that quantifies similarity between every pair of samples, transform that relation into a fuzzy equivalence relation, and then apply an alpha-cut to obtain crisp cluster membership. The resulting cluster structure is compared with the known benign and DDoS labels. Rather than producing only a cluster assignment, the framework uses the dominant within-class clusters and the inter-class overlap to estimate an adjusted accuracy and a corresponding false-alarm tendency [25].

3.2.1.2 CIC-DDoS2019 Dataset and Preprocessing

The experimental study uses the Canadian Institute for Cybersecurity DDoS 2019 (CIC-DDoS2019) dataset [27]. The subset employed in the paper contains approximately 225,000 labeled samples

consisting of benign and DDoS traffic. Each original sample contains 85 features, with the final feature providing the benign/DDoS label [25], [27].

The preprocessing stage removes nominal or discrete fields that are not suitable for the continuous fuzzy-similarity calculation, including flow identifier, source and destination Internet Protocol addresses, source and destination ports, and protocol. The timestamp is also excluded because its cyclical interpretation does not provide a direct continuous anomaly feature for the proposed analysis. The remaining continuous values span a wide numerical range, so each retained feature value is compressed to the interval from zero to one using the sigmoid transformation [25].

$$f(x_{im}) = \frac{1}{1 + e^{-x_{im}}} \tag{3.51}$$

where x_{im} denotes the mth retained feature of sample x_i. The transformation places the retained continuous features on a consistent bounded scale before pairwise fuzzy similarity is computed.

Table 3.7. CIC-DDoS2019 data configuration used for the fuzzy validation framework [25], [27].

Item	Configuration
Dataset	CIC-DDoS2019
Approximate samples used	225,000
Original features per sample	85, including class label
Classes	Benign and DDoS
Batch size	20,000 samples
Batch composition	Random balanced batch with equal benign and DDoS representation
Excluded fields	Flow ID, source/destination IP, source/destination port, protocol, and timestamp
Continuous-feature scaling	Sigmoid mapping to [0,1]

3.2.1.3 Fuzzy Tolerance Relation Construction

After preprocessing, the cosine amplitude method is used to construct the fuzzy tolerance relation [25], [28]. For a batch containing n samples, the pairwise similarities are collected in the $n \times n$ tolerance matrix

$$T = \begin{bmatrix} r_{11} & \cdots & r_{1n} \\ \vdots & \ddots & \vdots \\ r_{n1} & \cdots & r_{nn} \end{bmatrix} \tag{3.52}$$

The matrix is reflexive and symmetric. Each element r_{ij} quantifies the similarity between samples x_i and x_j. Following the notation used in the ARCI paper, the cosine-amplitude membership value is

$$r_{ij} = \frac{\left|\sum_{k=1}^{n} x_{ik}\, x_{jk}\right|}{\sqrt{\sum_{k=1}^{n} x_{ik}^2}\sqrt{\sum_{k=1}^{n} x_{jk}^2}} \tag{3.53}$$

The resulting membership value lies between zero and one. A value close to zero represents weak similarity, whereas a value close to one represents strong similarity. This pairwise representation preserves graded similarity rather than immediately forcing samples into crisp benign or DDoS groups [25].

3.2.1.4 Fuzzy Equivalence Relation Using Max-Min Composition

A fuzzy tolerance relation is reflexive and symmetric but is not necessarily transitive. The framework therefore converts the tolerance relation into a fuzzy equivalence relation through repeated max-min composition [25], [28]. For samples x_i and x_j, one composition step is expressed as

$$E(x_i, x_j) = \max_k \left[\min\left(r_{ik}, r_{kj}\right)\right] \tag{3.54}$$

where the intermediary sample x_k links x_i and x_j through the strongest available minimum similarity. The tolerance relation is composed with itself iteratively until successive relations no longer change. At convergence, the resulting fuzzy equivalence relation provides a stable relational representation from which clusters can be generated [25].

3.2.1.5 Defuzzification Using the α-Cut

The fuzzy equivalence relation is converted into a crisp relation using an α-cut. The alpha-cut specifies the minimum equivalence value that two samples must satisfy to remain linked within the same cluster. For a selected alpha in the interval $[0,1]$, the resulting crisp relation can be represented as

$$E_\alpha(x_i, x_j) = \begin{cases} 1, & E(x_i, x_j) \geq \alpha \\ 0, & E(x_i, x_j) < \alpha \end{cases} \tag{3.55}$$

Increasing alpha makes the membership requirement more restrictive. Low alpha values preserve many weak similarities and can allow substantial mixing between benign and DDoS samples. Higher values remove progressively weaker links, improving class separation but also discarding some within-class relationships. The experimental procedure therefore varies alpha rather than fixing it a priori and searches for a region that balances intra-class cohesion against inter-class ambiguity [25].

3.2.1.6 Batch Sampling and Clustering Procedure

Processing the complete 225,000-sample dataset as a single fuzzy relation would require an extremely large pairwise similarity matrix. To keep the analysis computationally manageable, the study randomly samples 20,000 observations at a time, with an equal representation of benign and DDoS traffic [25]. For each batch, the cosine-amplitude relation forms a 20,000 by 20,000 tolerance matrix, which is repeatedly composed using the max-min operation until the fuzzy relation stabilizes.

The stabilized fuzzy equivalence relation is then evaluated at a sequence of alpha-cut values. At each alpha, the crisp relation is inspected to determine how many samples remain strongly associated within the dominant DDoS and benign clusters and how many samples remain linked across the two classes. Repeating this process over randomly selected balanced batches allows the

validation framework to characterize the dataset rather than depend on one fixed ordering of samples [25].

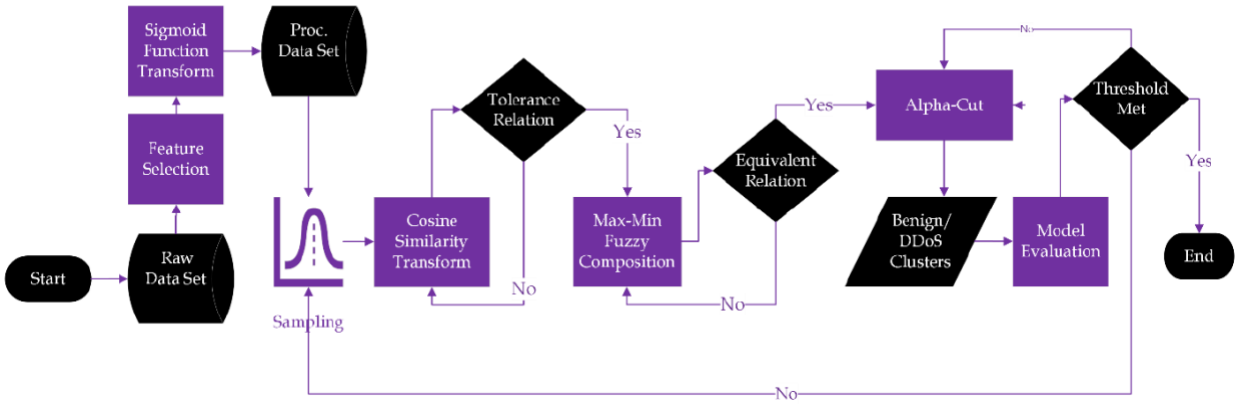


Figure 3.13. Fuzzy equivalence relation workflow used to estimate anomaly-detection accuracy. The process begins with feature selection and sigmoid normalization, repeatedly samples the dataset, constructs a cosine-similarity tolerance relation, applies max-min fuzzy composition until an equivalence relation is obtained, uses an alpha-cut to form benign and DDoS clusters, and evaluates the resulting model-validation criteria [25].

3.2.1.7 Intra-Cluster and Inter-Cluster Similarity Measures

The crisp equivalence matrix is partitioned according to the known class labels so that within-class cohesion and between-class mixing can be measured. In the 20,000-sample balanced batch, the paper associates the first 10,000 samples with one class and the second 10,000 with the other. The largest connected count within the DDoS region is denoted S_D, while the corresponding benign count is denoted S_B [25].

$$S_D = \text{maxcard}\{x_i : r_{ij} = 1,\ i,j \in [1,10000]\} \tag{3.56}$$

$$S_B = \text{maxcard}\{x_i : r_{ij} = 1,\ i,j \in [10001,20000]\} \tag{3.57}$$

Inter-class ambiguity is represented by the largest count of links crossing between the two class regions. The study refers to these ambiguous observations as tossed-up samples and denotes their count by S_T [25].

$$S_T = \text{maxcard}\{x_i : r_{ij} = 1,\ i \in [10001,20000],\ j \in [1,10000]\} \tag{3.58}$$

Large S_D and S_B values indicate strong internal class cohesion, while a large S_T indicates that benign and DDoS traffic remain difficult to separate at the selected alpha-cut. These quantities provide the basis for the validation metrics introduced next.

3.2.1.8 Adjusted Accuracy and False Alarm Rate

The fuzzy framework uses the tossed-up sample count to quantify ambiguity. The False Alarm Rate (FAR) used in the study is defined as the proportion of tossed-up samples relative to the batch size n [25]:

$$\mathrm{FAR} = \frac{S_T}{n} \tag{3.59}$$

To obtain a more conservative estimate of usable class separation, the paper introduces Adjusted Accuracy, denoted A_adj. This metric rewards the dominant within-class memberships represented by S_D and S_B while explicitly penalizing inter-class ambiguity S_T:

$$A_{\mathrm{adj}} = \frac{(S_D + S_B) - S_T}{n} \tag{3.60}$$

Adjusted Accuracy is therefore not simply the classification accuracy of a trained detector. It is a dataset-structure-based validation quantity derived from fuzzy cluster cohesion and overlap. The metric is evaluated across the alpha-cut range and used together with conventional confusion-matrix quantities to identify a practical operating region for performance benchmarking [25].

3.2.1.9 Experimental Evaluation Procedure

The evaluation sweeps the alpha-cut over increasingly restrictive similarity thresholds and records the resulting DDoS similarity, benign similarity, tossed-up similarity, false-alarm behavior, and adjusted-accuracy values. The ARCI study reports representative alpha-cuts from approximately 93.8% through 98.0% and examines confusion matrices at 95.0%, 95.5%, and 96.0% [25]. The objective of this sweep is to identify a balance between class separation and preservation of meaningful within-class structure, not simply to maximize one metric.

Quantitative findings, including the 95.5% alpha-cut operating point, the adjusted accuracy of 67.24%, and the expected peak accuracy of 89.57%, are reserved for Section 4.2.1. The present subsection establishes only the experimental design used to obtain those values. This separation keeps the methodology and results chapters consistent with the CARMEN+ final-report structure.

3.2.2 Contribution 2: Onset Localization Under Deceptive GNSS in Nonlinear State Estimators

GNSS spoofing can introduce counterfeit or manipulated navigation information that induces controlled position deviations while remaining plausible to the receiver [35]–[37]. Contribution 2 addresses deceptive GNSS attacks after an integrity monitor has already generated an alarm. Sequential detectors typically require multiple abnormal measurements before the accumulated evidence crosses a decision threshold, so the alarm time can occur after the true spoofing onset [47], [48], [52]. During this delay, corrupted measurements may already have propagated through a recursive navigation estimator. The onset-localization framework therefore uses the residual history preceding the alarm to estimate when the attack actually began [26].

The method is formulated as a retrospective change-point estimation problem in the residual sequence of a nonlinear state estimator. It develops a finite-window detectability analysis, distinguishes the onset-identifiability properties of step and ramp spoofing, derives a correlation-aware generalized likelihood ratio localizer, and compares that model-based approach with a retrospective cumulative-sum baseline and a lightweight temporal neural localizer [26], [31]–[33].

3.2.2.1 Problem Formulation as a Change-Point Estimation Problem

At time index k, let z_k denote the m-dimensional measurement vector and let $\hat{z}_{k|k-1}$ denote the measurement predicted by the nonlinear navigation estimator. The innovation or residual is defined as [26]

$$\nu_k = z_k - \hat{z}_{k|k-1} \tag{3.61}$$

with residual covariance S_k. To place residual components on a comparable statistical scale, the residual is whitened using its covariance:

$$y_k = S_k^{-1/2}\nu_k \tag{3.62}$$

Let k_{alarm} denote the index at which an external integrity monitor declares spoofing. For a retrospective window of w normalized residuals ending at the alarm, the residuals are stacked as

$$y = \left[y_{k_{\text{alarm}}-w+1}^{\top} \quad \cdots \quad y_{k_{\text{alarm}}}^{\top}\right]^{\top} \tag{3.63}$$

The stacked vector is modeled using a Gaussian mean-shift representation

$$y \sim \mathcal{N}\left(\mu_{k_{\text{onset}}}, \Sigma\right) \tag{3.64}$$

where Σ is the covariance of the stacked window and μ_j is the mean pattern associated with a hypothesized onset index j. The candidate set $\mathcal{C}$ contains the possible onset indices within the alarm-conditioned window. Onset localization is therefore the problem of selecting the candidate j whose mean-shift model best explains the observed residual window [26], [32].

Table 3.8. Key notation used in the GNSS spoofing onset-localization formulation [26].

Symbol	Description
ν_k	Navigation-filter residual at sample k
S_k	Residual covariance at sample k
y_k	Normalized residual $S_k^{(-1/2)}\nu_k$
y	Stacked normalized residual window ending at k_{alarm}
Σ	Covariance of the stacked residual window
$\mathcal{C}$	Candidate onset indices within the alarm-conditioned window
μ_j	Mean residual-window pattern for candidate onset j
k_{onset}	True spoofing onset index
k_{alarm}	Spoofing alarm index produced by the integrity monitor

3.2.2.2 Nonlinear State Estimator and Residual Sequence Construction

The residual sequence is generated by an Unscented Kalman Filter (UKF), which is appropriate for the nonlinear bicycle-kinematic motion model used in the experimental study [26], [30]. At every 10-Hz GNSS update, the UKF predicts the measurement, computes the residual v_k and its covariance S_k, and produces the normalized residual y_k in Equation (3.62). The same normalized residual stream is supplied to all onset localizers so that performance differences arise from the localization method rather than from different residual-generation pipelines.

For the enhanced neural localizer, the experimental implementation also records internal estimator diagnostics: the Kalman gain K_k, the Normalized Innovation Squared (NIS), and the state covariance P_k. These quantities are not required by the CUSUM or C-GLR localizers, but they provide additional information about how the navigation filter is responding to the evolving measurement inconsistency [26].

3.2.2.3 Finite-Window Spoofing Detectability

For each candidate onset j, the finite-window problem is first considered as a binary test between nominal and spoofed residual behavior [26]:

$$H_0: y \sim \mathcal{N}(0, \Sigma), \qquad H_1(j): y \sim \mathcal{N}(\mu_j, \Sigma) \tag{3.65}$$

The statistical separability of the two hypotheses is governed by the covariance-weighted residual spoof energy

$$E = \mu_j^\top \Sigma^{-1} \mu_j \tag{3.66}$$

This quantity is the squared Mahalanobis magnitude of the mean shift and acts as an effective signal-to-noise measure after temporal covariance is taken into account. Under the Gaussian mean-shift model and a prescribed false-alarm probability α, the Neyman-Pearson detection probability is [26], [33]

$$P_D^\star(\alpha) = 1 - \Phi\left(\Phi^{-1}(1-\alpha) - \sqrt{E}\right) \tag{3.67}$$

where Φ(.) is the standard normal cumulative distribution function. If β denotes the allowable miss probability, reliable finite-window detection with P_D at least 1-β requires

$$E \geq [\Phi^{-1}(1-\alpha) + \Phi^{-1}(1-\beta)]^2 \tag{3.68}$$

The detectability condition provides a theoretical explanation for the experimental trend observed later in Chapter 4: when spoof energy is low relative to the residual covariance, onset hypotheses are difficult to distinguish regardless of the localization algorithm.

3.2.2.4 Step and Ramp Spoofing Identifiability

Detectability of an attacked window does not guarantee precise localization of the change point. The temporal structure of the attack determines how distinct neighboring onset hypotheses are. To isolate this effect, the paper considers the simplified case $\Sigma = I$. For a step bias of magnitude b that changes immediately at the onset, the accumulated residual energy over w samples is [26]

$$E_{\text{step}}(w) = wb^2 \tag{3.69}$$

Because the step profile changes abruptly, shifting the hypothesized onset by even one sample changes the mean pattern immediately. Neighboring onset candidates are therefore comparatively well separated.

For a ramp attack with slope b-dot and sampling interval Δt, the residual shift grows gradually from zero and its energy is

$$E_{\mathrm{ramp}}(w) = \dot{b}^2 \Delta t^2 \frac{(w-1)w(2w-1)}{6} \tag{3.70}$$

Although the total ramp energy grows rapidly with window length, the earliest post-onset residuals remain small. Consequently, mean profiles corresponding to adjacent onset hypotheses overlap strongly near the true onset. The paper therefore predicts, and later demonstrates experimentally, that ramp spoofing is intrinsically more difficult to localize precisely than an abrupt step bias [26].

3.2.2.5 Correlated Generalized Likelihood Ratio Onset Localizer

To incorporate both temporal covariance and the expected spoofing profile, the candidate mean shift is represented as a parametric subspace

$$\mu_j = A_j \theta \tag{3.71}$$

where A_j is a deterministic step or ramp template aligned with candidate onset j and θ contains the unknown bias parameters. Maximizing the Gaussian likelihood with respect to θ yields the correlated Generalized Likelihood Ratio (C-GLR) statistic [26], [32]:

$$\Lambda(j) = y^{\top} \Sigma^{-1} A_j \left(A_j^{\top} \Sigma^{-1} A_j \right)^{-1} A_j^{\top} \Sigma^{-1} y \tag{3.72}$$

The onset estimate is selected as the candidate with the largest likelihood statistic

$$\hat{k}_{\mathrm{onset}} = \underset{j \in \mathcal{C}}{\mathrm{argmax}} \Lambda(j) \tag{3.73}$$

For each alarm instance, the implementation evaluates every candidate in C using both the residual covariance and the selected attack template. The reported configuration uses step and ramp templates with covariance regularization $\lambda = 10^{-4}$ to improve numerical stability [26].

3.2.2.6 Retrospective CUSUM Baseline

A retrospective Cumulative Sum (CUSUM) localizer is included as the conventional evidence-accumulation baseline [26], [31]. The scalar residual-energy sequence is $l_k = y_k^T y_k$. A one-sided CUSUM state accumulates sustained positive deviation above the nominal reference level k_pos according to

$$g_k = \max\left(0, g_{k-1} + \ell_k - k_{\mathrm{pos}}\right), \qquad \ell_k = y_k^{\top} y_k \tag{3.74}$$

After an alarm, the onset is estimated retrospectively as the earliest candidate whose forward CUSUM segment increases monotonically toward k_alarm:

$$\hat{k}_{\mathrm{onset}}^{\mathrm{CUSUM}} = \min\left\{ j \in \mathcal{C} : g_j \le g_{j+1} \le \cdots \le g_{k_{\mathrm{alarm}}} \right\} \tag{3.75}$$

The experimental baseline uses $k_{pos} = 0.5$, threshold $h = 8.0$, and a minimum run length of five samples [26]. This method does not explicitly model step or ramp signatures or the full temporal covariance and therefore provides a useful reference for quantifying the benefit of the more structured C-GLR and neural approaches.

Table 3.9. Model-based onset-localizer configuration [26].

Parameter	Value
CUSUM drift/reference level k_{pos}	0.5
CUSUM threshold h	8.0
Minimum run length	5 samples
C-GLR templates	Step and ramp
Covariance regularization λ	10^{-4}

3.2.2.7 Neural Temporal Onset Localizer

The learning-based localizer treats onset estimation as a sequence-to-label problem. The parity feature vector at time k combines the normalized residual, residual energy, and first-order residual change [26]:

$$x_k = [y_k^\top \quad \| y_k \|^2 \quad (y_k - y_{k-1})^\top]^\top \tag{3.76}$$

For m residual components, the feature dimension is $d = 2m + 1$. The features within the alarm-conditioned window are stacked as

$$X = [x_{k_{\text{alarm}}-w+1} \quad \cdots \quad x_{k_{\text{alarm}}}]^\top \in \mathbb{R}^{w \times d} \tag{3.77}$$

A lightweight one-dimensional temporal convolutional network maps X to one score for each onset candidate. The scores are converted to a categorical probability distribution by the softmax function

$$p_\psi(j|X) = \frac{e^{s_j}}{\sum_{i \in \mathcal{C}} e^{s_i}} \tag{3.78}$$

and the onset estimate is the candidate with maximum probability

$$\hat{k}_{\text{onset}}^{\text{NN}} = \underset{j \in \mathcal{C}}{\text{argmax}} p_\psi(j|X) \tag{3.79}$$

The model parameters ψ are learned using categorical negative log likelihood

$$\mathcal{L}(\psi) = -\log p_\psi(j^\star|X) \tag{3.80}$$

Two feature configurations are evaluated. The parity configuration uses only the residual-derived features in Equation (3.76). The enhanced configuration augments these inputs with UKF diagnostics, including K_k, NIS, and the trace of P_k. This allows the experiment to test whether

internal estimator behavior contains onset information not captured by the residual sequence alone [26].

Table 3.10. Neural onset-localizer configuration [26].

Parameter	Value
Architecture	Conv1D classifier
Convolution blocks	2
Kernel size	3
Hidden width	128
Dropout	0.1
Optimizer	Adam
Learning rate	10^{-3}
Training epochs	30

3.2.2.8 KITTI-Derived Dataset and Spoofing Scenarios

The onset-localization experiments use GNSS measurement sequences derived from the KITTI driving dataset [26], [29]. Each realization contains a 30-s trajectory segment sampled at 10 Hz, giving $N = 300$measurements. Nominal GNSS error is modeled as zero-mean Gaussian noise with covariance $R = 0.25I_2$, corresponding to a standard deviation of 0.5 m per axis [26].

Spoofing is injected synthetically beginning at a randomly selected onset index. Monte Carlo sampling varies the onset location, detector delay, step-bias magnitude, and ramp slope so that the localizers are evaluated over a broad range of attack timing and strength. The distributions used by the study are summarized in Table 3.11.

Table 3.11. Monte Carlo spoofing parameter distributions used in the onset-localization study [26].

Parameter	Distribution	Description
k_{onset}	U[0.12N, 0.88N]	Attack onset index
d	U[5, 40] samples	Detection delay
b	U[3, 10] m	Step attack bias magnitude
$\dot{b}$	U[0.1, 0.8] m/s	Ramp attack slope

3.2.2.9 Unscented Kalman Filter Residual Generation

All three onset-localization approaches use the same UKF-generated residual stream. The UKF employs bicycle kinematics with a 2.71-m wheelbase and standard scaled sigma-point parameters. The measurement covariance matches the simulated GNSS noise model, and the initial and process covariances are specified explicitly in the paper so that the residual sequence can be reproduced [26], [30].

Table 3.12. UKF configuration used for residual-stream generation [26], [30].

Parameter	Value
Motion model	Bicycle kinematics
Wheelbase L	2.71 m
Sigma-point parameters (α, β, κ)	$(10^{-3}, 2, 0)$
Measurement covariance R	$0.25I_2$
Initial state $\mathbf{s}_0$	**0**
Initial covariance P_0	$diag(0.05^2, 0.05^2, (0.5°)^2, 0.05^2)$
Process noise Q	$diag(0.05^2, 0.05^2, (0.5°)^2, 0.8^2)$

The normalization in Equation (3.62) is applied at every sample using the corresponding residual covariance S_k. This step is essential because the subsequent statistical formulation assumes that the residual sequence is evaluated relative to the estimator-predicted uncertainty rather than through raw measurement differences alone.

3.2.2.10 Alarm-Conditioned Candidate Window and Evaluation Metrics

All onset estimators operate retrospectively on the same candidate window ending at k_{alarm}. The window length is $w = 50$ samples, equivalent to 5 s at the 10-Hz sampling rate [26]. This interval exceeds the maximum simulated detector delay of 40 samples, ensuring that the true attack onset remains within the search window for the generated realizations and that sufficient post-onset information is available for distinguishing step and ramp signatures.

Localization accuracy is measured from the absolute difference between the estimated and true onset indices

$$e_{\mathrm{on}} = \sqrt{\left(\hat{k}_{\mathrm{onset}} - k_{\mathrm{onset}}\right)^2} \tag{3.81}$$

The study summarizes this error using the empirical cumulative distribution function, median absolute onset error, Mean Absolute Error (MAE), and Root Mean Square Error (RMSE). It also reports the probabilities that the onset estimates fall within one or two samples of the ground truth:

$$P_1 = P(e_{\text{on}} \leq 1), \qquad P_2 = P(e_{\text{on}} \leq 2) \tag{3.82}$$

At 10 Hz, these correspond to tolerances of ±0.1 s and ±0.2 s, respectively. Performance is additionally evaluated as a function of the spoof energy in Equation (3.66) and separately for step and ramp attack profiles. The numerical localization results are presented in Section 4.2.2; this chapter is limited to the analytical formulation, estimator configuration, data generation, and evaluation protocol used to obtain those results [26].

4. Results and Findings

4.1 Subtask 1: Detection and Identification

The results for Subtask 1 are presented in the same chronological order as the technical contributions described in Section 3.1. The first contribution evaluates the data-driven displacement-based Global Navigation Satellite System (GNSS) spoofing detector published at the 2025 IEEE 4th International Conference on AI in Cybersecurity (ICAIC) [10]. The results are analyzed with respect to attack-specific detection behavior, nominal residual bounds, and the analytical motion-consistency baseline introduced earlier in the project.

For this contribution, performance is reported using accuracy, recall, and F1 score. The underlying experiments use real-world driving data from the Honda Research Institute Driving Dataset (HDD) [14] with synthetically injected GNSS spoofing attacks, as described in Section 3.1.1. The discussion below emphasizes both the numerical detection performance and the practical effect of replacing analytical position reconstruction with short-horizon data-driven displacement prediction.

4.1.1 Contribution 1: ICAIC Data-Driven Detection Results

The ICAIC study demonstrates that the data-driven displacement predictor preserves the strong detection capability of the analytical framework for conventional spoofing attacks while substantially tightening the nominal motion-error boundary [10]. Under clean driving conditions, the maximum data-driven displacement residual is 1.79 m, whereas the analytical position-reconstruction framework requires a 3.35 m threshold. The 1.56 m reduction corresponds to approximately 46.6% and directly increases sensitivity to smaller GNSS-induced inconsistencies.

This threshold reduction is the central result that distinguishes the data-driven framework from the analytical baseline. The improvement is obtained without using GNSS as an input to the displacement predictor. Instead, the neural network learns the relationship between trusted in-vehicle speed, heading information, and short-horizon displacement. Consequently, the GNSS measurement remains an independent information source against which the predicted vehicle motion is tested.

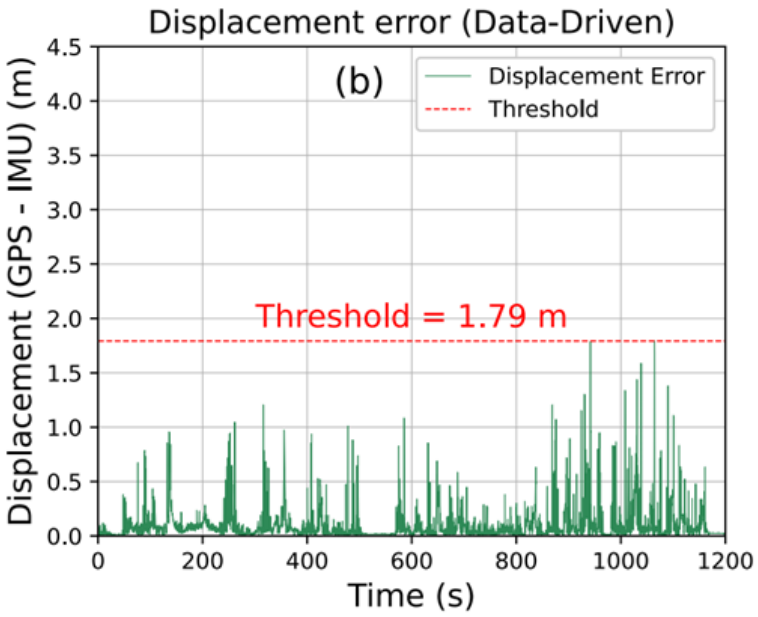


Figure 4.1. Data-driven displacement residual under clean driving conditions and the extracted 1.79 m detection threshold. The comparatively narrow nominal residual band provides greater sensitivity to spoofing-induced displacement inconsistencies.

The ICAIC results are organized below by spoofing-attack type, followed by a direct comparison with the analytical baseline.

4.1.1.1 Turn-by-Turn Attack Detection

Turn-by-turn spoofing attacks were evaluated by introducing controlled deviations into the GNSS-reported trajectory while the physical vehicle remained in motion. Because the spoofed trajectory was generated to remain smooth and kinematically plausible, the attack did not rely on an abrupt change in reported speed. Detection therefore depended primarily on the disagreement between GNSS-derived displacement and the displacement predicted from trusted onboard measurements [10].

A representative result is shown in Figure 4.2. Before attack onset, the displacement residual remains below the 1.79 m nominal threshold. After the spoofing manipulation begins, the residual rises above the threshold and the attack is identified. The lower data-driven threshold allows the trajectory inconsistency to be recognized earlier than with the 3.35 m analytical position threshold, while the use of onboard speed in the classification logic confirms that the vehicle itself remains in forward motion.

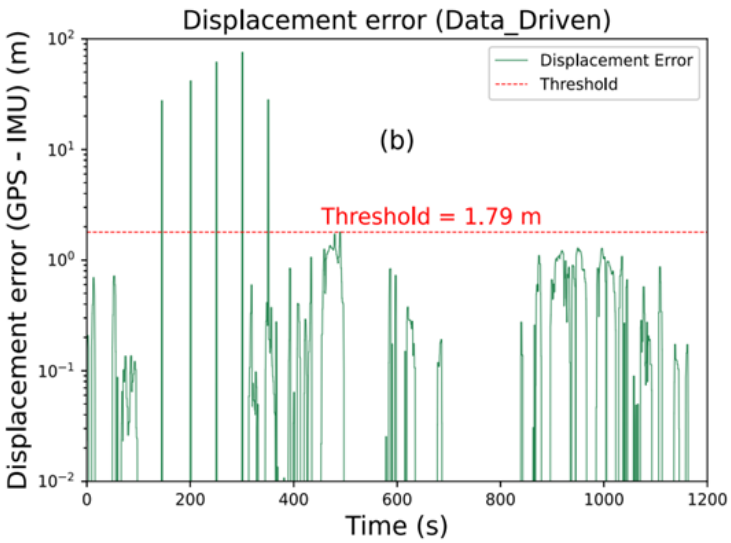


Figure 4.2. Representative turn-by-turn spoofing detection using the data-driven displacement framework. The displacement residual exceeds the 1.79 m threshold after attack onset, enabling timely detection.

Across the evaluated turn-by-turn scenarios, the data-driven detector achieved 100.00% accuracy, 100.00% recall, and a 100.00% F1 score [10]. These values indicate that all evaluated attack samples were detected without sacrificing the overall classification performance. Relative to the analytical framework, which achieved 99.74% accuracy for the same attack category, the improvement in accuracy is modest; the more important benefit is the lower residual threshold, which provides additional margin for detecting subtler trajectory deviations.

4.1.1.2 Stop Attack Detection

Stop attacks were evaluated under conditions in which the physical vehicle was stationary while the manipulated GNSS information indicated motion. This attack produces a direct inconsistency between the onboard speed measurement and GNSS-derived speed, so the speed residual provides the dominant detection cue. The data-driven displacement estimate supplies an additional motion-consistency check and preserves the same interpretable attack-identification logic used by the analytical method.

The data-driven framework achieved 99.95% overall accuracy, 95.23% recall, and a 97.56% F1 score for stop attacks [10]. The high accuracy confirms that the detector remained reliable over the complete evaluated sequence; however, the recall is lower than the corresponding accuracy and is also lower than the recall obtained for the other conventional attack types. This indicates that a small subset of stop-attack samples was not identified at the correct decision instant, even though the overall classification remained highly accurate.

This result is important when interpreting the contribution as a whole. The data-driven formulation was not introduced because the analytical detector failed on obvious speed-inconsistency attacks. In fact, the analytical method already performed extremely well for stop attacks. The principal value of the learned displacement model is that it retains nearly the same performance for these

conventional attacks while creating a substantially tighter displacement-error boundary for more subtle trajectory manipulation.

4.1.1.3 Overshoot Attack Detection

Overshoot attacks represent the complementary speed-inconsistency condition: the physical vehicle continues moving while the spoofed GNSS signal indicates a stationary or near-stationary state. As with stop attacks, the attack produces a pronounced mismatch between GNSS-derived and onboard vehicle speed. The decision logic therefore uses the speed residual together with the GNSS speed state to distinguish overshoot behavior from other spoofing categories.

The data-driven detector achieved 99.91% accuracy, 99.54% recall, and a 99.77% F1 score for overshoot attacks [10]. These results are effectively near-perfect and show that replacing analytical position reconstruction with learned displacement prediction does not degrade the detector's ability to recognize attacks characterized by explicit speed disagreement.

The small difference between the 99.91% data-driven accuracy and the 100.00% analytical accuracy is not practically meaningful in the context of the contribution. Both frameworks are highly effective for overshoot attacks. The key conclusion is that the data-driven method extends sensitivity toward smaller displacement manipulations while preserving the strong conventional-attack performance already provided by the analytical baseline.

4.1.1.4 Multiple Small Biased Attack Detection

Multiple small biased attacks provide the most informative test of the data-driven contribution. Rather than introducing one abrupt displacement jump, the attacker applies repeated small GNSS biases. Each individual manipulation is approximately 1.41 m, which remains below the 1.79 m conventional maximum-residual threshold used by the data-driven detector and well below the 3.35 m analytical position threshold. Nevertheless, the repeated biases generate a large accumulated localization deviation [10].

Figure 4.3 illustrates the distinction between nominal behavior, a single large-magnitude attack, and a sequence of small biased manipulations. Under nominal conditions, GNSS and onboard displacement remain closely aligned. A single large attack produces an obvious deviation. In the multiple-small-biased case, the individual points remain much closer to nominal behavior, but the repeated bias creates a systematic divergence. This pattern demonstrates why a detector based only on a conservative analytical threshold can fail to recognize the early stages of a gradual attack.

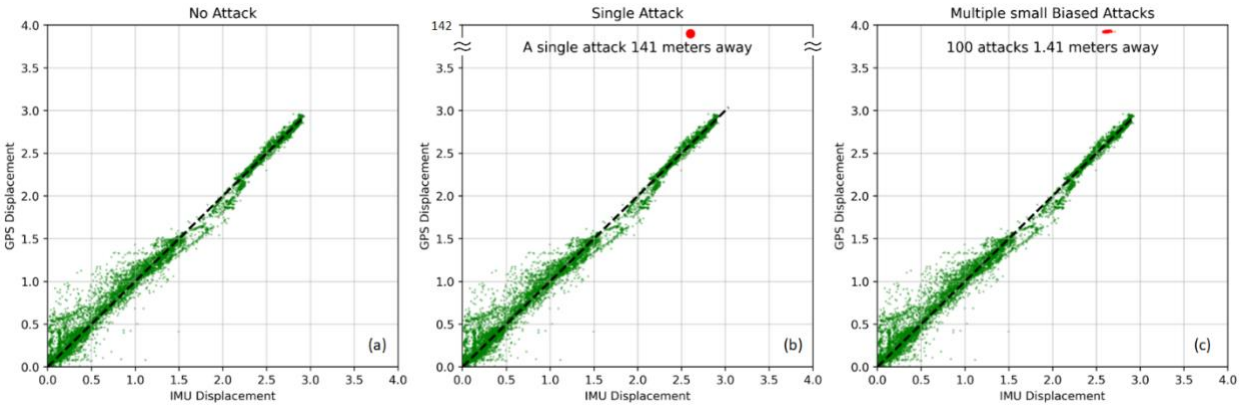


Figure 4.3. Comparison of GNSS and onboard displacement under three conditions: no attack, a single large-magnitude spoofing attack, and multiple small biased attacks. Small individual biases can remain close to the nominal relationship while accumulating into a substantial localization error.

To improve sensitivity to this condition, the clean data-driven residual distribution is characterized statistically and a three-standard-deviation criterion is used for the multiple-small-biased scenario. The resulting detector achieved 97.83% accuracy, 100.00% recall, and a 98.90% F1 score [10]. The 100.00% recall is particularly important because it indicates that the evaluated attack samples were not missed. Since recall is perfect, the difference between 100% and the reported overall accuracy is associated with false-positive classifications rather than undetected attack samples.

These results establish the principal advantage of the first data-driven contribution: learning displacement directly from onboard measurements reduces nominal residual uncertainty enough to expose gradual spoofing steps that can remain hidden beneath an analytical position-error threshold. At the same time, the result also reveals the remaining limitation of a purely threshold-based strategy. A sophisticated attacker can design smaller or slower biases that remain within even the tighter statistical boundary for extended periods. This limitation motivates the adaptive and time-series methods presented in Contributions 2 and 3.

4.1.1.5 Analytical versus Data-Driven Performance

Table 4.1 summarizes the attack-specific performance of the data-driven detector. The results show perfect or near-perfect accuracy for the three conventional spoofing categories and strong performance for multiple small biased attacks. The stop-attack recall of 95.23% is the lowest recall among the four categories, whereas the multiple-small-biased detector reaches 100.00% recall despite the much more subtle attack construction.

Table 4.1. Data-driven GNSS spoofing detection performance reported for the ICAIC contribution [10].

Attack Type	Accuracy (%)	Recall (%)	F1 Score (%)
Turn-by-turn	100.00	100.00	100.00
Stop	99.95	95.23	97.56
Overshoot	99.91	99.54	99.77
Multiple small biased	97.83	100.00	98.90

A direct comparison with the analytical baseline clarifies where the data-driven method provides an actual performance gain. The analytical detector already achieves 99.74%, 99.99%, and 100.00% accuracy for turn-by-turn, stop, and overshoot attacks, respectively. The corresponding data-driven values are 100.00%, 99.95%, and 99.91%. Thus, for conventional large-magnitude attacks, the two frameworks should be regarded as comparably effective rather than interpreting the data-driven method as uniformly superior on every accuracy metric.

Table 4.2. Comparison of the analytical and data-driven ICAIC detection frameworks [10].

Comparison Item	Analytical Framework	Data-Driven Framework	Result
Nominal displacement/position threshold	3.35 m	1.79 m	46.6% lower threshold
Turn-by-turn accuracy	99.74%	100.00%	+0.26 percentage points
Stop accuracy	99.99%	99.95%	-0.04 percentage points
Overshoot accuracy	100.00%	99.91%	-0.09 percentage points
Multiple small biased attacks	NA	97.83% accuracy; 100.00% recall	Data-driven sensitivity extended to gradual attacks

The decisive improvement is therefore not a large increase in conventional-attack accuracy; it is the reduction of the nominal residual threshold from 3.35 m to 1.79 m. This approximately 46.6% reduction increases the detector's sensitivity to smaller GNSS trajectory deviations while preserving essentially the same performance for obvious stop, overshoot, and turn-by-turn attacks. The learned displacement representation also avoids the cumulative integration error inherent in analytical position propagation, which is the main reason a tighter residual boundary can be maintained.

Taken together, the results demonstrate a useful division of strengths. The analytical formulation remains attractive because of its direct physical interpretation and strong performance for large, structured spoofing attacks. The data-driven displacement formulation maintains that capability while providing a substantially tighter uncertainty boundary and a practical mechanism for detecting multiple small biased attacks. The remaining sensitivity of any fixed statistical boundary to changing nominal conditions and increasingly stealthy temporal attack patterns provides the motivation for the adaptive DBSCAN and time-series contributions evaluated in Sections 4.1.2 and 4.1.3.

4.1.2 Contribution 2: Adaptive DBSCAN Results

The second contribution evaluates the adaptive Density-Based Spatial Clustering of Applications with Noise (DBSCAN) framework published in the 2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC) [11]. The evaluation retained the data-driven displacement predictor from Contribution 1 and changed the below-threshold anomaly-detection stage so that the DBSCAN neighborhood radius could evolve with the recursively estimated statistics of nominal displacement residuals. Five subsets of the Honda Research Institute Driving Dataset (HDD) [14] were used, and each attack category was evaluated through 25 independently generated attack realizations.

The results are interpreted in two parts. Conventional turn-by-turn, stop, and overshoot attacks provide a check that adding the adaptive clustering branch does not weaken the strong detection

capability already obtained for structured attacks. Multiple small biased attacks provide the more demanding test because the individual 1.41 m spoofing steps remain below the 1.79 m conventional displacement threshold and must therefore be identified from their inconsistency with the evolving nominal residual distribution.

4.1.2.1 Adaptive Threshold Behavior

The adaptive detector was initialized from 120,000 clean residual samples and used a dynamic neighborhood radius defined as $\varepsilon = 5\sigma$, where σ was updated recursively from samples accepted as non-anomalous [11]. During online operation, a residual that remained within the current adaptive boundary was incorporated into the recursive mean and standard-deviation estimates, whereas a residual classified as anomalous was excluded from subsequent updates. This selective-update mechanism allowed the nominal cluster to follow legitimate changes in residual dispersion without permitting identified attack samples to pull the boundary toward spoofed behavior.

The practical significance of this behavior is that the detector is not tied to a single error magnitude established during offline calibration. If nominal residual variability increases because of changing vehicle dynamics or sensor conditions, the adaptive radius can expand; if nominal behavior becomes tighter, the radius can contract. The published SMC study reports the resulting attack-detection performance rather than a separate time history of ε, so no unsupported numerical range for the evolving radius is introduced here. The evidence for the usefulness of adaptation is therefore assessed through the detector's repeated-trial sensitivity, specificity, and area under the receiver operating characteristic curve (AUC) across the four attack categories.

4.1.2.2 Conventional Spoofing Attack Detection

For conventional spoofing attacks, the adaptive framework preserved high detection performance. Turn-by-turn attacks were detected with an average accuracy of 98.62 ± 1.0%, 100% sensitivity, 98.62 ± 1.1% specificity, and an AUC of 99.3 ± 0.6% [11]. The perfect sensitivity indicates that the evaluated attack samples were identified, while the slightly lower specificity reflects a small number of normal samples being classified as anomalous.

Stop attacks achieved 99.96 ± 0.1% accuracy, 98.9 ± 1.8% sensitivity, 100% specificity, and 99.44 ± 0.9% AUC [11]. Overshoot attacks achieved 99.88 ± 0.1% accuracy, 99.16 ± 0.2% sensitivity, 100% specificity, and 99.63 ± 0.5% AUC [11]. These results show that the adaptive branch does not materially compromise the detection of attacks that already produce clear speed or displacement inconsistencies. In particular, the 100% specificity reported for stop and overshoot attacks indicates strong resistance to false alarms in the evaluated normal samples.

The conventional-attack results should not be interpreted as evidence that adaptive DBSCAN is required for these attack types. The earlier fixed-threshold framework already performed very strongly for turn-by-turn, stop, and overshoot attacks. Instead, these experiments serve as an important compatibility test: the addition of recursive statistical adaptation preserves high conventional-attack performance while creating a second mechanism for detecting residual patterns that remain below the fixed threshold.

4.1.2.3 Multiple Small Biased Attack Detection

The primary benefit of the adaptive framework is observed for multiple small biased spoofing attacks. In the SMC experiment, the GNSS coordinates were shifted by 1 m in both planar directions

for each attack step, producing an approximately 1.41 m Euclidean displacement change. Because 1.41 m remains below the 1.79 m conventional displacement threshold, the individual manipulations are intentionally difficult to identify using a fixed pointwise threshold. The attack was repeated five times on each of five clean HDD subsets, yielding 25 independent realizations [11].

For this attack category, the adaptive DBSCAN detector achieved 98.38 ± 0.1% accuracy, 100% sensitivity, 99.18 ± 1.1% specificity, and an AUC of 99.3 ± 0.6% [11]. The 100% sensitivity is particularly significant because it shows that the evaluated gradual attacks were detected despite the fact that each individual spoofing step remained below the conventional system threshold. At the same time, the 99.18% specificity indicates that the increased sensitivity was not obtained by indiscriminately labeling nominal residuals as attacks.

This result demonstrates the value of monitoring the evolving statistical structure of the displacement residual rather than relying only on whether a single sample exceeds a fixed magnitude. A sequence of small biases can alter the density and location of the residual distribution even when the individual samples remain close to the nominal range. The adaptive clustering formulation is therefore better matched to gradual changes in error behavior than a purely static boundary.

4.1.2.4 Detection Performance and Robustness

Table 4.3 summarizes the published SMC results. All values are reported as percentages, and the mean and standard deviation reflect repeated attack realizations across the selected HDD subsets [11].

Table 4.3. Adaptive DBSCAN GNSS spoofing detection performance reported in the IEEE SMC 2025 contribution [11].

Attack Type	Accuracy (%)	Sensitivity (%)	Specificity (%)	AUC (%)
Turn-by-turn	98.62 ± 1.0	100	98.62 ± 1.1	99.3 ± 0.6
Stop	99.96 ± 0.1	98.9 ± 1.8	100	99.44 ± 0.9
Overshoot	99.88 ± 0.1	99.16 ± 0.2	100	99.63 ± 0.5
Multiple small biased	98.38 ± 0.1	100	99.18 ± 1.1	99.3 ± 0.6

Across all four attack categories, accuracy remains above 98%, while AUC remains at or above 99.3% [11]. The repeated-trial results therefore indicate that the adaptive detector provides a stable operating region across different attack timings and vehicle-motion conditions. The conventional attacks retain high performance, and the gradual multiple-small-biased scenario achieves perfect sensitivity with high specificity.

The robustness improvement should be understood primarily in terms of the decision boundary rather than a dramatic increase in conventional-attack accuracy. The adaptive detector continuously updates the statistics of accepted nominal residuals using constant-memory recursive calculations, which allows the boundary to accommodate non-stationary error behavior without batch re-clustering. At the same time, anomalous samples are excluded from the updates, reducing the risk that a detected spoofing sequence will progressively redefine itself as normal.

These characteristics address an important limitation of the fixed-threshold framework and motivate the subsequent use of explicit temporal memory for even smaller and slower spoofing patterns.

4.1.3 Contribution 3: Time-Series Detection Results

The third contribution evaluates the time-series framework published in the IEEE Open Journal of Vehicular Technology [12]. The framework combines the established data-driven displacement predictor with a dedicated attack-probability predictor that analyzes recent displacement-error history. Conventional turn-by-turn, stop, and overshoot attacks continue to be handled by the displacement- and speed-consistency branch, while multiple small biased attacks are evaluated using 25 s and 50 s temporal windows and a fixed 50% attack-probability decision threshold.

The published evaluation uses five HDD test subsets and repeats each attack scenario 25 times [12], enabling the mean and variability of accuracy, F1 score, sensitivity, specificity, and AUC to be assessed. Particular emphasis is placed on the multiple-small-biased scenarios because the individual spoofing steps are reduced to approximately 0.56 m and 0.28 m, corresponding to approximately 3σ and 1.5σ of the nominal GNSS-versus-onboard displacement error, respectively.

4.1.3.1 Conventional Spoofing Attack Performance

The combined time-series framework retained strong performance for conventional attacks. The data-driven branch achieved 98.62 ± 1.0% accuracy, 99.28 ± 0.6% F1 score, 100% sensitivity, 98.62 ± 1.1% specificity, and 99.3 ± 0.6% AUC for turn-by-turn attacks [12]. Stop attacks achieved 99.96 ± 0.1% accuracy, 99.42 ± 0.9% F1 score, 98.9 ± 1.8% sensitivity, 100% specificity, and 99.44 ± 0.9% AUC. Overshoot attacks achieved 99.88 ± 0.1% accuracy, 99.5 ± 0.17% F1 score, 99.16 ± 0.2% sensitivity, 100% specificity, and 99.63 ± 0.5% AUC [12].

These values confirm that introducing the temporal attack-prediction stage does not require sacrificing the established performance for attacks that create immediate motion inconsistency. In the overall architecture, the time-series component is complementary rather than a replacement for the conventional branch: abrupt attacks are still detected from their direct effect on displacement and speed residuals, whereas the temporal predictor is reserved for below-threshold patterns that require recent-history information.

4.1.3.2 Multiple Small Biased Attack Performance

Two increasingly stealthy multiple-small-biased scenarios were evaluated. For n = 250, the attack was distributed over 250 processed samples, corresponding to approximately 25 s, with an individual Euclidean spoofing step of approximately 0.56 m. This magnitude is approximately three standard deviations of the nominal displacement residual and remains substantially below the 1.79 m conventional threshold [12]. The n = 250 configuration achieved 95.92 ± 1.7% accuracy, 95.84 ± 3.1% F1 score, 93.86 ± 7.1% sensitivity, 95.88 ± 1.7% specificity, and 95.82 ± 3.0% AUC.

For n = 500, the same approximate 141 m accumulated localization shift was distributed across 500 processed samples, corresponding to approximately 50 s. The individual spoofing step was reduced to approximately 0.28 m, or approximately 1.5σ of the nominal displacement error [12]. Despite the substantially smaller per-step manipulation, the detector achieved 93.66 ± 5.6% accuracy, 92.98 ± 5.7% F1 score, 92.44 ± 6.7% sensitivity, 93.7 ± 5.7% specificity, and 93.04 ± 5.7% AUC.

The reduction in mean performance and the larger standard deviations for n = 500 are consistent with the greater difficulty of the 1.5σ attack. The attack steps lie much closer to nominal sensor uncertainty, so the temporal signature must become sufficiently persistent before the predicted probability exceeds the fixed 50% decision threshold. Even under this more difficult condition, the framework demonstrates that attacks with individual steps as small as approximately 0.28 m can be identified using recent-history information.

4.1.3.3 Effect of Time-Series Window Length

The two temporal configurations provide different recent-memory horizons. The n = 250 model uses approximately 25 s of residual history, while the n = 500 model uses approximately 50 s [12]. The shorter window is more responsive to a faster accumulation of larger spoofing steps, whereas the longer window provides additional temporal context for a slower attack whose individual deviations remain closer to nominal uncertainty.

In the reported experiments, n = 250 produced higher mean accuracy, F1 score, sensitivity, specificity, and AUC than n = 500. However, the effect of window length cannot be isolated from these experiments alone because the attack magnitude was changed at the same time: the 25 s case used 0.56 m steps, whereas the 50 s case used 0.28 m steps. Therefore, the lower n = 500 performance should not be attributed solely to the longer window. It reflects the combined effect of a longer temporal horizon and a substantially more stealthy 1.5σ attack.

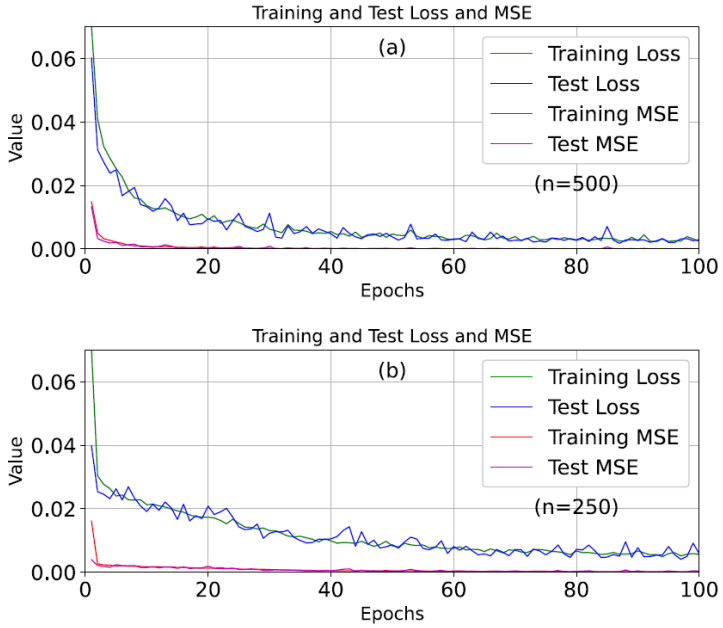


Figure 4.4. Training and testing behavior of the attack-probability predictor for the n = 500 and n = 250 temporal configurations [12]. Both models converge over 100 epochs without a pronounced separation between training and testing curves.

The training behavior shown in Figure 4.4 supports the feasibility of maintaining separate predictors for the two temporal scales. Both configurations exhibit smooth convergence with closely aligned training and testing curves. The figure does not, by itself, establish which temporal window is

intrinsically optimal; selecting an operating window should instead be tied to the expected rate and magnitude of spoofing evolution.

4.1.3.4 Effect of Attack Magnitude

Attack magnitude has a direct effect on detectability because the temporal predictor must distinguish a persistent malicious pattern from normal displacement-error variability. The n = 250 scenario uses approximately 0.56 m steps, corresponding to about 3σ, while the n = 500 scenario reduces the step size to approximately 0.28 m, or about 1.5σ [12]. Both scenarios accumulate to approximately 141 m of location shift, demonstrating that a large long-term navigation error can be produced even when each individual manipulation is much smaller than the conventional 1.79 m detection threshold.

The approximately twofold reduction in per-step attack magnitude is accompanied by a decrease in mean accuracy from 95.92% to 93.66% and in mean F1 score from 95.84% to 92.98%. Sensitivity similarly decreases from 93.86% to 92.44%, while AUC decreases from 95.82% to 93.04% [12]. This trend is expected because the 0.28 m deviations overlap more strongly with nominal uncertainty and therefore provide less instantaneous evidence of an attack.

As with the window-length analysis, attack magnitude and temporal horizon are coupled in the experimental design. The results demonstrate that the framework remains effective as the attack becomes smaller and slower, but they do not provide a controlled one-variable-at-a-time estimate of sensitivity to attack magnitude. A future ablation in which step magnitude is varied while holding the time-series window fixed would be required to quantify that relationship independently.

4.1.3.5 Detection Latency

Temporal averaging introduces an inherent delay between attack onset and the corresponding change in the averaged differential-error signal. The time-series framework addresses this issue through a latency-compensated target feature derived from the moving mean of the differential error. In the published n = 500 example, the uncompensated differential-error representation visibly lags the actual attack label, while the compensated target is aligned so that its transition crosses the 0.5 decision level near the attack transition [12].

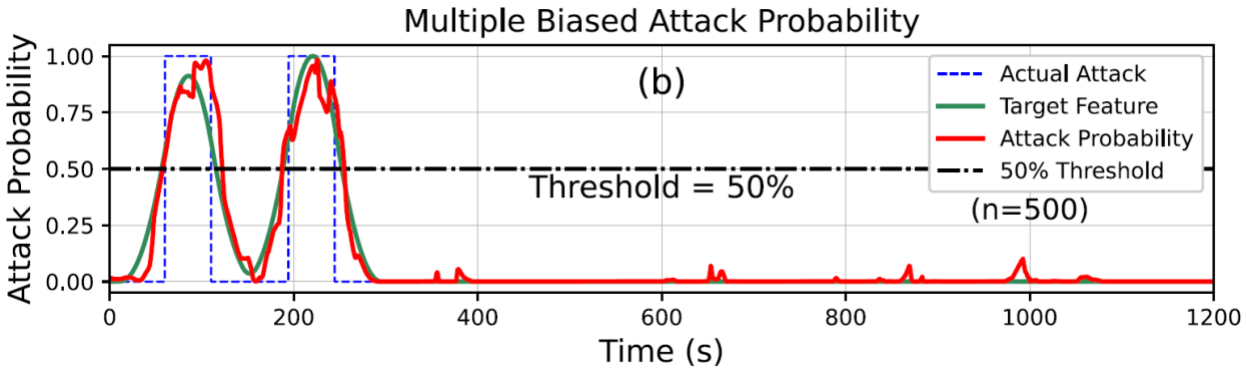


Figure 4.5. Latency compensation and real-time attack-probability prediction for the n = 500 multiple-small-biased scenario [12]. The compensated target reduces the lag introduced by

moving-average feature construction, and attack probability is converted to a binary decision using a fixed 50% threshold.

The predicted probability is compared directly with a 50% threshold: values above 0.5 are classified as spoofing and values at or below 0.5 are treated as normal [12]. This hard decision rule avoids adding a second confirmation window that would further delay the response. The paper demonstrates the latency-compensation behavior graphically and describes the resulting detector as operating in real time; however, it does not report a separate scalar statistic such as mean detection latency in seconds across all 25 repetitions. Accordingly, no numerical latency value is introduced in this report.

4.1.3.6 Overall Performance

Table 4.4 reproduces the principal published performance results for the complete data-driven framework. The conventional attack rows summarize the displacement- and speed-consistency branch, while the two multiple-small-biased rows summarize the time-series attack-probability predictor [12].

Table 4.4. Time-series GNSS spoofing detection performance reported in the IEEE Open Journal of Vehicular Technology contribution [12].

Attack Type	Accuracy (%)	F1 Score (%)	Sensitivity (%)	Specificity (%)	AUC (%)
Turn-by-turn	98.62 ± 1.0	99.28 ± 0.6	100	98.62 ± 1.1	99.3 ± 0.6
Stop	99.96 ± 0.1	99.42 ± 0.9	98.9 ± 1.8	100	99.44 ± 0.9
Overshoot	99.88 ± 0.1	99.5 ± 0.17	99.16 ± 0.2	100	99.63 ± 0.5
Multiple biased (n = 250)	95.92 ± 1.7	95.84 ± 3.1	93.86 ± 7.1	95.88 ± 1.7	95.82 ± 3.0
Multiple biased (n = 500)	93.66 ± 5.6	92.98 ± 5.7	92.44 ± 6.7	93.7 ± 5.7	93.04 ± 5.7

Table 4.5 isolates the two gradual-attack configurations to clarify the relationship among recent-memory duration, per-step spoofing magnitude, and detection performance. Because the same approximate 141 m cumulative localization shift is distributed over different numbers of attack steps, the experiment directly demonstrates that cumulative consequence alone does not determine detectability; the instantaneous attack magnitude and the rate at which the bias evolves are also critical.

Table 4.5. Comparison of the two multiple-small-biased time-series configurations [12].

Window n	Recent Memory	Attack Step	Statistical Magnitude	Accuracy (%)	F1 Score (%)
250	25 s	0.56 m	≈ 3σ	95.92 ± 1.7	95.84 ± 3.1
500	50 s	0.28 m	≈ 1.5σ	93.66 ± 5.6	92.98 ± 5.7

Overall, the time-series contribution extends the detectable attack regime well below the conventional 1.79 m displacement threshold. Conventional turn-by-turn, stop, and overshoot attacks remain near 99% accuracy, while the temporal predictor identifies gradual spoofing steps down to approximately 0.28 m, or 1.5σ of the nominal displacement residual [12]. The lower performance and larger variability in the n = 500 case appropriately reflect the increased difficulty of distinguishing such small biases from normal uncertainty.

The results also define the practical role of the time-series method within the layered detection architecture. Instantaneous displacement and speed checks remain efficient for large structured attacks; adaptive DBSCAN addresses changes in the nominal residual distribution; and the temporal probability predictor provides a recent-memory mechanism for attacks whose individual steps are too small to reliably trigger either static or adaptive pointwise decisions. This progression is consistent with the project objective of progressively addressing more stealthy GNSS spoofing behaviors without abandoning the interpretable detection mechanisms developed in the earlier contributions.

4.1.4 Contribution 4: Stereo Visual Navigation Results

The fourth contribution reports the current experimental results of the ongoing stereo visual navigation (VN) study for short-horizon planar global displacement prediction in Global Navigation Satellite System (GNSS)-compromised and GNSS-denied conditions [23]. In contrast to Contributions 1–3, this work is not yet a peer-reviewed publication. Accordingly, the numerical values reported in this subsection are treated as demonstrated results from the current manuscript and controlled experimental testbed, while hardware deployment, broader validation, and publication remain ongoing activities.

The evaluation follows the methodology documented in Section 3.1.4. The same 101-dimensional geometric representation is used for independent direction classification of dx_g and dy_g and for four direction-conditioned XGBoost displacement predictors [21], [23]. Classification performance is evaluated on held-out validation and test data, displacement prediction is evaluated using mean absolute error (MAE), root mean squared error (RMSE), and coefficient of determination (R^2), and recursively accumulated visual displacements are evaluated at the trajectory level using absolute trajectory error (ATE) and translational relative pose error (RPE) [23].

4.1.4.1 Direction Classification Results

The sign-classification stage is evaluated only outside the ±5 mm transition region defined in Section 3.1.4.7. This exclusion avoids assigning a hard positive or negative direction to extremely small motions that are comparable to feature-localization, stereo-matching, disparity, and synchronization uncertainty. The remaining samples are divided using the 70% training, 15% validation, and 15% test split described in the methodology [23].

For longitudinal displacement, the dx_g classifier achieved 88.19% validation accuracy and 88.50% test accuracy. The corresponding test sensitivity, specificity, F1 score, and area under the receiver operating characteristic curve (AUC) were 88.06%, 88.92%, 88.22%, and 95.50%, respectively. For lateral displacement, the dy_g classifier achieved 88.33% validation accuracy and 88.00% test accuracy, with test sensitivity of 87.56%, specificity of 88.46%, F1 score of 88.21%, and AUC of 95.29% [23]. The close agreement between validation and test results indicates that the classifiers

generalize consistently to unseen samples rather than exhibiting a large validation-to-test degradation.

Table 4.6. Performance of the current VN displacement sign classifiers [23]. All values are reported in percent.

Classifier	Accuracy	Sensitivity	Specificity	F1 Score	AUC
dx_g Validation	88.19	87.97	88.39	88.02	95.00
dx_g Test	88.50	88.06	88.92	88.22	95.50
dy_g Validation	88.33	88.45	88.21	88.36	95.30
dy_g Test	88.00	87.56	88.46	88.21	95.29

The test confusion matrices provide a complementary view of the classification behavior. For dx_g, the matrix contains approximately 45.5% true-negative samples and 43.0% true-positive samples, with 5.8% false negatives and 5.7% false positives. For dy_g, the corresponding proportions are approximately 43.1% true negatives, 44.9% true positives, 6.4% false negatives, and 5.6% false positives [23]. The relatively balanced false-positive and false-negative fractions are consistent with the approximately balanced sensitivity and specificity reported in Table 4.6.

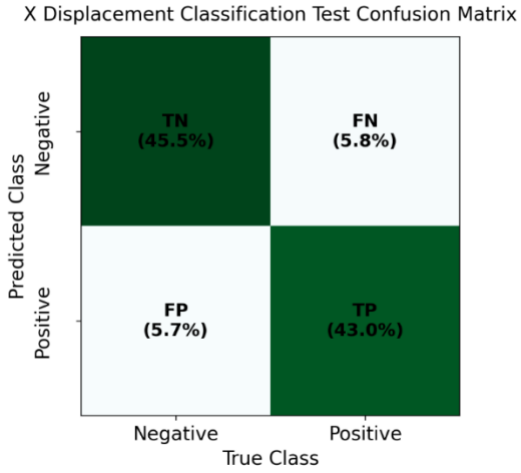


Figure 4.6. Test confusion matrix for the global longitudinal displacement sign classifier, dx_g [23].

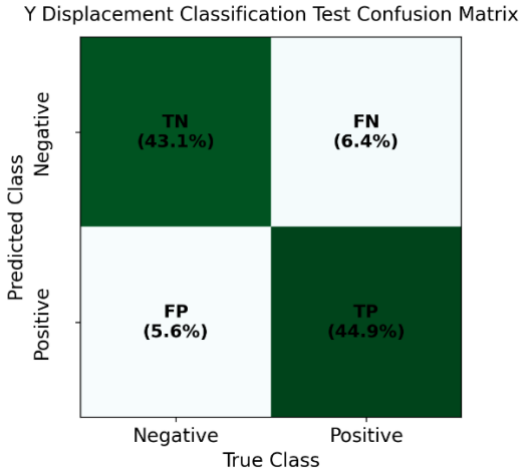


Figure 4.7. Test confusion matrix for the global lateral displacement sign classifier, dy_g [23].

4.1.4.2 Global dx_g Prediction Results

Following sign classification, separate XGBoost regressors are applied to negative and positive dx_g samples. The negative dx_g predictor achieved a test MAE of 4.7914 mm, test RMSE of 6.6537 mm, and test R^2 of 0.8393. The positive dx_g predictor achieved a slightly lower test MAE of 4.4592 mm and RMSE of 6.3640 mm, together with a higher test R^2 of 0.8619 [23]. The corresponding validation metrics remain close to the test values, which supports consistent model behavior on held-out data.

Table 4.7. Validation and test performance of the four direction-conditioned XGBoost displacement predictors in the current VN manuscript [23].

Model	MAE (mm)	RMSE (mm)	R^2
Negative dx_g (Validation)	4.8115	6.7080	0.8370
Negative dx_g (Test)	4.7914	6.6537	0.8393
Positive dx_g (Validation)	4.4658	6.3387	0.8671
Positive dx_g (Test)	4.4592	6.3640	0.8619
Negative dy_g (Validation)	5.0378	6.9205	0.7436
Negative dy_g (Test)	4.9193	6.8539	0.7455
Positive dy_g (Validation)	4.7655	6.6353	0.7597
Positive dy_g (Test)	4.5599	6.3929	0.7746

The test scatter plot shows that the predicted longitudinal displacements follow the ideal one-to-one relationship across both negative and positive motion. Although residual spread remains around the reference line, the direction-conditioned estimates remain concentrated around the ground-truth trend over the approximately ±70 mm displacement range represented in the current experiment [23]. This behavior is consistent with the R^2 values above 0.83 for both longitudinal predictors.

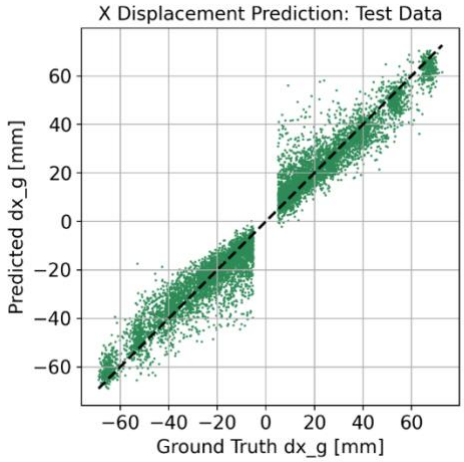


Figure 4.8. Test global longitudinal displacement prediction results. The dashed line represents the ideal one-to-one relationship between predicted and ground-truth dx_g [23].

4.1.4.3 Global dy_g Prediction Results

The lateral displacement component is predicted using separate negative and positive dy_g regressors. The negative dy_g model achieved a test MAE of 4.9193 mm, RMSE of 6.8539 mm, and R^2 of 0.7455. The positive dy_g model achieved a test MAE of 4.5599 mm, RMSE of 6.3929 mm, and R^2 of 0.7746 [23]. As with the longitudinal predictors, validation and test errors remain similar, indicating that the reported test behavior is not the result of a pronounced validation-to-test performance drop.

The lower dy_g R^2 values relative to the corresponding dx_g values indicate that lateral displacement remains the more difficult prediction problem in the current dataset. The test scatter plot nevertheless follows the one-to-one reference trend across negative and positive displacement values. The larger spread around the reference line is consistent with the lower R^2 values and suggests greater visual ambiguity in the lateral mapping from image-space motion and stereo disparity to global displacement [23].

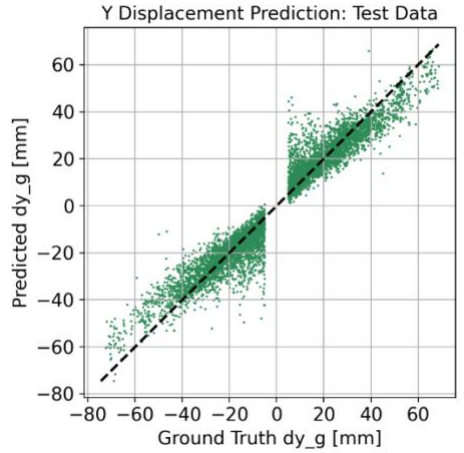


Figure 4.9. Test global lateral displacement prediction results. The dashed line represents the ideal one-to-one relationship between predicted and ground-truth dy_g [23].

4.1.4.4 Prediction Error Distributions

The displacement-error distributions provide a direct measure of bias and spread after the direction-conditioned prediction stage. For dx_g, the test error distribution is centered near zero with a mean error of -0.05 mm and a standard deviation of 6.51 mm. The corresponding three-standard-deviation bound is 19.54 mm, which is approximately 8.68% of the 225 mm experimental vehicle length [23]. The near-zero mean indicates negligible systematic longitudinal bias in the current test data.

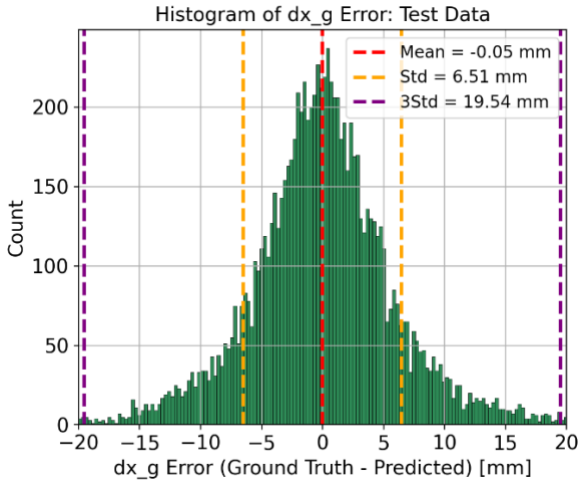


Figure 4.10. Distribution of test longitudinal displacement errors for the positive and negative dx_g prediction models [23].

For dy_g, the mean test error is 0.06 mm and the standard deviation is 6.62 mm, producing a three-standard-deviation bound of 19.86 mm, or approximately 8.83% of the vehicle length [23]. The small mean again indicates negligible systematic bias. The lateral standard deviation is only slightly larger than the longitudinal value, even though the dy_g regression R^2 values are lower, indicating that both axes maintain similar absolute error scales over the current operating range.

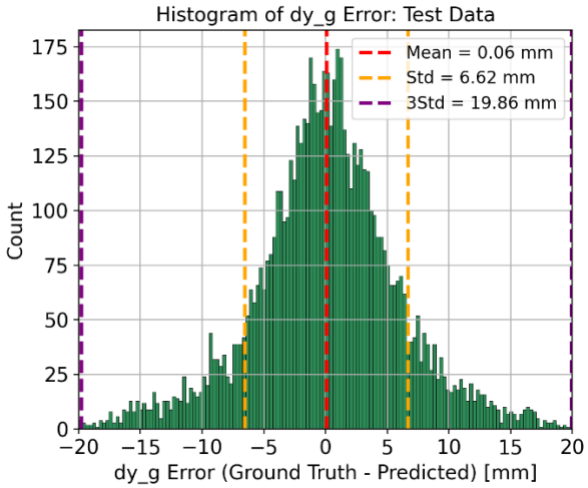


Figure 4.11. Distribution of test lateral displacement errors for the positive and negative dy_g prediction models [23].

The direction-conditioned architecture provides a substantial improvement over the single-model baseline that does not separate positive and negative motion. The single dx_g model produced an error standard deviation of 18.54 mm and a three-standard-deviation bound of 55.62 mm, whereas the conditional dx_g framework reduced the bound to 19.54 mm, an approximately 64.9% reduction. The corresponding single dy_g baseline produced a 16.62 mm standard deviation and a 49.85 mm three-standard-deviation bound; the conditional dy_g framework reduced the bound to 19.86 mm, an approximately 60.2% reduction [23].

Table 4.8. Effect of direction-conditioned prediction on the test displacement-error distributions [23].

Axis	Single-model σ (mm)	Single-model 3σ (mm)	Conditional σ (mm)	Conditional 3σ (mm)	3σ Reduction
dx_g	18.54	55.62	6.51	19.54	64.9%
dy_g	16.62	49.85	6.62	19.86	60.2%

4.1.4.5 Trajectory-Level Evaluation

Trajectory-level behavior is evaluated by recursively accumulating predicted planar displacement increments from a known starting position. The resulting visual trajectory is compared with the VICON ground-truth trajectory and with a classical Kabsch rigid-alignment baseline combined with Random Sample Consensus (RANSAC) outlier rejection [22], [24]. To maintain a consistent comparison, both approaches use the same SIFT correspondences, chained KLT tracks, and available ten-point stereo-temporal quadruples [23].

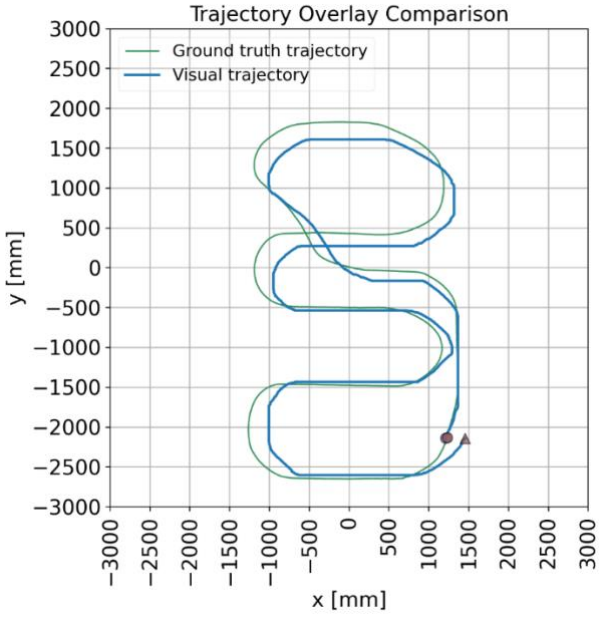


Figure 4.12. Overlay comparison of the VICON ground-truth trajectory and the trajectory reconstructed from the proposed visual displacement predictor [23].

The proposed visual predictor achieved an ATE RMSE of 210.92 mm and a translational RPE RMSE of 7.12 mm. Under the same correspondence inputs, the Kabsch+RANSAC baseline produced an ATE RMSE of 899.40 mm and a translational RPE RMSE of 38.77 mm [23], [24]. Relative to this baseline, the proposed framework reduces ATE RMSE by approximately 76.5% and translational RPE RMSE by approximately 81.6%. These reductions support the value of the engineered 101-feature representation and conditional classification-prediction structure for converting sparse stereo-temporal observations into short-horizon global displacement increments.

Table 4.9. Trajectory reconstruction performance comparison for the current VN study [23], [24].

Method	**ATE RMSE (mm)**	**Translational RPE RMSE (mm)**
Visual predictor	210.92	7.12
Kabsch+RANSAC baseline	899.40	38.77

The two trajectory metrics characterize different aspects of performance. The 7.12 mm RPE RMSE reflects the local consistency of predicted displacement increments, whereas the larger 210.92 mm ATE RMSE reflects error accumulation after many increments are recursively integrated. The ATE result therefore shows that the current visual predictor is effective for short-horizon motion estimation but still exhibits accumulated trajectory drift. This distinction is important when considering the method as a future input to a higher-level visual-inertial or multisensor localization architecture rather than as a completed standalone global localization solution.

4.1.4.6 Computational Performance

The current VN manuscript does not report measured stage-level or end-to-end execution times for the complete stereo processing and prediction pipeline [23]. Therefore, a numerical real-time performance claim is not made in this results section. The demonstrated results concern classification accuracy, displacement prediction error, and trajectory reconstruction; computational timing remains to be experimentally quantified on the intended deployment hardware.

Stereo images were recorded at 20 Hz during the experiments, corresponding to a 50 ms interval between synchronized stereo frames [23]. This 50 ms interval provides a useful throughput target for a future implementation that is expected to produce one new estimate per incoming stereo pair. However, the Raspberry Pi 4 documented in the experimental platform was used for data acquisition, and the manuscript does not demonstrate that the complete SIFT, chained KLT, stereo association, 101-feature construction, classification, and XGBoost regression pipeline executes within 50 ms on that device.

Several design choices are favorable for eventual edge implementation: only ten stereo-temporal quadruples are retained, the learned representation contains 101 scalar features, and XGBoost inference avoids recurrent-state propagation and iterative nonlinear trajectory optimization [21], [23]. Nevertheless, SIFT extraction, stereo matching, geometric filtering, and chained KLT tracking

remain nontrivial image-processing stages. A defensible computational-performance result will require average, median, and worst-case timing for each stage and for the complete pipeline, together with processor, software, and parallelization details. Until those measurements are available, real-time implementation is treated as an anticipated capability rather than a demonstrated result.

4.1.4.7 Current Limitations

The current results establish the feasibility of the proposed stereo geometric representation and direction-conditioned displacement prediction framework, but the scope of the demonstrated evidence remains limited. First, the experiments were conducted using a 225 mm differential-drive vehicle in a controlled indoor miniature city. Although the individual algorithmic stages are not intrinsically tied to vehicle scale, full-scale outdoor operation has not yet been validated under changes in illumination, weather, road texture, camera vibration, longer feature distances, and more complex traffic scenes [23].

Second, VICON provides millimeter-level global ground truth during dataset generation and evaluation. VICON is not required during inference, but the current study has not yet replaced the controlled motion-capture reference with an outdoor ground-truth system. Third, the sign classifiers exclude the ±5 mm transition region around zero displacement; consequently, the reported direction-classification metrics should not be interpreted as validated performance for arbitrarily small vehicle motions. Fourth, recursively accumulating short-horizon predictions produces nonzero trajectory drift, as reflected by the 210.92 mm ATE RMSE even though the local translational RPE RMSE is 7.12 mm [23].

Fifth, the present framework is visual-only. Integration with the recorded Inertial Measurement Unit (IMU) data to form a Visual-Inertial Navigation System (VINS) remains future work. The current manuscript also identifies dynamic-object handling as an extension: an object-detection stage such as You Only Look Once (YOLO) is planned to favor stationary image regions before stereo-temporal matching so that features on independently moving objects have less influence on motion prediction [23]. Finally, end-to-end runtime on an embedded target has not yet been measured, and the VN manuscript remains in preparation rather than peer-reviewed publication. The current CARMEN+ report therefore distinguishes the experimental results summarized above from anticipated full-scale deployment, real-time edge implementation, visual-inertial integration, dynamic-object filtering, and eventual journal publication.

Table 4.10. Demonstrated and ongoing elements of the current stereo VN contribution [23].

Element	Status	Current Evidence / Remaining Work
Direction classification	Demonstrated	Held-out test accuracies of 88.50% for dx_g and 88.00% for dy_g outside the ±5 mm transition region.
Short-horizon displacement prediction	Demonstrated	Direction-conditioned XGBoost predictors evaluated on held-out test data with approximately 6.4–6.9 mm RMSE.

Element	Status	Current Evidence / Remaining Work
Trajectory reconstruction	Demonstrated	ATE RMSE of 210.92 mm and translational RPE RMSE of 7.12 mm in the controlled indoor experiment.
Full-scale outdoor validation	Ongoing	Not yet demonstrated; requires outdoor trajectories, full-scale camera geometry, and independent ground truth.
Real-time edge execution	Ongoing	20 Hz acquisition provides a 50 ms throughput target, but end-to-end pipeline timing has not yet been reported.
VINS integration / dynamic-object filtering / publication	Ongoing	IMU fusion, stationary-region filtering, and peer-reviewed journal publication are planned future extensions.

4.2 Subtask 2: Data-Driven Verification and Validation of Attacked HATS

The results for Subtask 2 evaluate two complementary verification and validation (V&V) contributions. Contribution 1 examines whether the structure of a labeled cyber-traffic dataset supports a defensible performance benchmark for data-driven anomaly-detection models. Contribution 2 evaluates whether the true onset of a deceptive Global Navigation Satellite System (GNSS) attack can be reconstructed after an integrity monitor raises an alarm. The first contribution therefore concerns validation of expected model performance, whereas the second concerns post-alarm validation of when corrupted navigation information began to affect a nonlinear estimator [25], [26].

The results are reported using the evaluation procedures defined in Section 3.2. For the fuzzy-equivalence framework, emphasis is placed on the effect of the α-cut on within-class cohesion, between-class ambiguity, Adjusted Accuracy, false-alarm behavior, and the estimated peak accuracy of a detector trained on the same data [25]. For onset localization, the four evaluated methods are the retrospective Cumulative Sum (CUSUM) baseline, the correlated Generalized Likelihood Ratio (C-GLR) localizer, a neural localizer using residual-derived parity features, and an enhanced neural localizer that also uses internal Unscented Kalman Filter (UKF) diagnostics [26].

4.2.1 Contribution 1: Fuzzy Equivalence Relation Validation Results

The fuzzy-equivalence contribution was evaluated on approximately 225,000 samples from the CIC-DDoS2019 dataset [25], [27]. The framework repeatedly sampled balanced 20,000-sample batches and varied the α-cut applied to the fuzzy equivalence relation. The resulting cluster structure was then used to quantify how strongly Distributed Denial of Service (DDoS) and benign samples remained associated within their own classes and how much ambiguity persisted between the two classes. The reported results are therefore properties of the dataset and validation framework, not the test accuracy of a newly trained DDoS classifier.

4.2.1.1 Effect of α-Cut on Cluster Separation

The α-cut controls the minimum fuzzy-equivalence value required for two samples to remain associated. At lower α-cut values, many sample pairs satisfy the similarity requirement, so both within-class cohesion and cross-class association remain high. As the α-cut is increased, weaker relationships are removed and the two classes become more distinguishable; however, the same stricter criterion also removes legitimate within-class associations. The paper therefore identifies an explicit tradeoff between improved class separation and loss of cluster cohesion [25].

Table 4.11 focuses on the operating region surrounding the selected α-cut. Between 95.0% and 96.0%, the DDoS similarity measure decreases from 9.60×10^3 to 7.40×10^3, while the benign similarity measure decreases from 9.60×10^3 to 8.58×10^3. Over the same interval, the tossed-up similarity decreases from 5.95×10^3 to 4.04×10^3 [25]. The reduction in tossed-up similarity indicates less ambiguity between the two classes, but the simultaneous decline in the two within-class measures shows the cost of enforcing greater separation.

Table 4.11. Selected fuzzy-equivalence similarity results near the identified α-cut operating point [25].

α-Cut (%)	DDoS Similarity ($\times10^3$)	Benign Similarity ($\times10^3$)	Tossed-Up Similarity ($\times10^3$)	Total ($\times10^3$)
95.0	9.60	9.60	5.95	25.15
95.2	9.04	9.50	5.18	23.72
95.5	8.53	9.38	4.47	22.38
95.7	7.66	9.17	4.27	21.10
96.0	7.40	8.58	4.04	20.01

The trend confirms that there is no single α-cut that maximizes every desirable property. A very permissive threshold preserves large clusters but also retains substantial overlap between benign and DDoS samples. A very restrictive threshold reduces cross-class ambiguity but progressively fragments the class structure. The validation problem is therefore to identify an operating point that balances these competing effects rather than maximizing only one similarity measure [25].

4.2.1.2 Intra-Cluster and Inter-Cluster Similarity Results

The experimental analysis separates the equivalence matrix into regions representing DDoS-to-DDoS similarity, benign-to-benign similarity, and cross-class association. The DDoS and benign measures characterize peak intra-cluster cohesion, whereas the tossed-up measure captures inter-cluster ambiguity. Across the α-cut sweep, all three measures decrease, but their relative rates determine whether a candidate threshold improves class distinction enough to justify the accompanying loss of within-class structure [25].

At the selected 95.5% α-cut, the reported DDoS similarity is 8.53×10^3, benign similarity is 9.38×10^3, and tossed-up similarity is 4.47×10^3 [25]. Relative to the 95.0% condition, the tossed-up measure is reduced by approximately 24.9%, while the DDoS and benign measures decrease by approximately 11.1% and 2.3%, respectively. This local comparison illustrates why 95.5% is a useful compromise in the reported experiment: cross-class ambiguity is reduced substantially while a comparatively large portion of the within-class structure is retained.

The paper further constructs illustrative confusion matrices at α-cuts of 95.0%, 95.5%, and 96.0% under the stated assumption that samples missing from the maximum cluster are treated as misclassified. These matrices are reproduced in Table 4.12 to show how stricter α-cut selection changes the balance of assignments. The entries are retained exactly as reported in the source study and are used here as part of the paper's validation analysis rather than as results from an independently trained classifier [25].

Table 4.12. Confusion-matrix constructions reported for selected α-cut values in the fuzzy-equivalence study [25].

α-Cut (%)	Ground-Truth Class	Predicted DDoS	Predicted Benign
95.0	DDoS	9,604	405
95.0	Benign	396	9,595
95.5	DDoS	8,532	619
95.5	Benign	1,468	9,381
96.0	DDoS	7,398	1,422
96.0	Benign	2,602	8,578

4.2.1.3 Adjusted Accuracy and False Alarm Tradeoff

The principal validation metric introduced by the fuzzy study is Adjusted Accuracy, which rewards the dominant DDoS and benign clusters while penalizing tossed-up samples. This formulation is intended to prevent a high apparent accuracy from being interpreted as sufficient evidence of reliability when considerable ambiguity remains near the class boundary [25].

As the α-cut increases, the false-alarm-related tossed-up component decreases because fewer weak cross-class relationships survive. At the same time, increasingly strict membership requirements reduce the number of samples retained within the dominant DDoS and benign clusters. The performance curves reported in the paper therefore do not improve monotonically with α-cut. Instead, the curves reveal a region in which class separation has improved enough to reduce ambiguity but before the loss of intra-class cohesion becomes dominant [25].

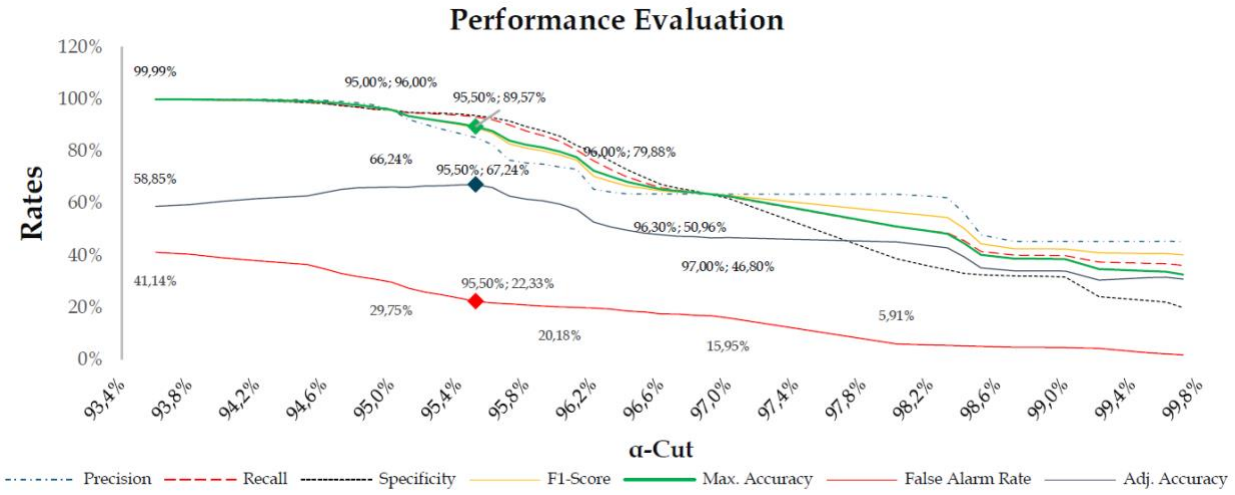


Figure 4.13. Performance evaluation across α-cut values for the fuzzy-equivalence validation framework [25]. The selected 95.5% operating point balances the reduction of cross-class ambiguity against the loss of within-class cohesion.

Figure 4.13 is particularly important because it shows why the selected α-cut should not be interpreted as the point of highest possible value for every individual metric. Model reliability is treated as a balance among class separation, false alarms, and retained cluster structure. This is consistent with the V&V purpose of the contribution: the objective is to establish a defensible benchmark for expected detector performance rather than to optimize a classifier directly.

4.2.1.4 Peak Accuracy Estimation

Through heuristic adjustment of the α-cut, the study identifies 95.5% as the reported operating point. At this setting, Adjusted Accuracy reaches 67.24% and the estimated maximum or peak accuracy is 89.57% [25]. These values are the central quantitative result of the fuzzy-equivalence contribution.

The distinction between the two values is important. The 89.57% value represents the framework's estimate of the peak accuracy that a data-driven DDoS model could be expected to achieve under the dataset structure captured at the selected α-cut, whereas the 67.24% Adjusted Accuracy applies the additional penalty associated with ambiguous or tossed-up samples. Accordingly, the lower Adjusted Accuracy provides a more conservative view of class distinguishability, while the peak value provides an upper performance reference for model validation [25].

These values are specific to the CIC-DDoS2019 data, preprocessing, sampling procedure, and fuzzy-equivalence construction used in the study. They should therefore not be interpreted as universal accuracy limits for DDoS detection. Their value is methodological: they demonstrate how a dataset-derived benchmark can be established before or alongside training so that a subsequently reported detector accuracy can be judged against an independent structural reference [25], [27].

4.2.1.5 Validation Implications for Data-Driven Anomaly Detection

The results demonstrate that dataset structure can provide useful information about the performance that should reasonably be expected from a data-driven anomaly detector. A high post-training accuracy value alone does not reveal whether the classes were intrinsically easy to separate, whether substantial overlap remains, or whether the reported result is consistent with the separability of the underlying data. The fuzzy-equivalence framework adds this missing validation layer by quantifying similarity before accepting detector performance at face value [25].

For HATS cybersecurity, the significance is broader than the specific DDoS case used for demonstration. Safety-critical deployment requires evidence that a detection model is reliable under the data conditions for which it is intended. The reported results show one way to derive a dataset-specific benchmark that combines expected accuracy with ambiguity and false-alarm considerations. However, the present evidence is limited to CIC-DDoS2019 and binary benign-versus-DDoS traffic. The paper explicitly identifies extension to other cyber threats as future work [25]. Consequently, this contribution establishes a V&V methodology and a demonstrated DDoS case study, but does not yet establish that the same numerical benchmark transfers directly to other HATS attack modalities.

4.2.2 Contribution 2: GNSS Spoofing Onset Localization Results

The second contribution evaluates post-alarm localization of deceptive GNSS attack onset using 18,034 independent held-out spoofing realizations derived from KITTI trajectories [26], [29]. The experiments compare four estimators: retrospective CUSUM, C-GLR, a neural onset localizer using parity residual features, and an enhanced neural localizer that augments those features with UKF diagnostics. Performance is reported using median absolute onset error, mean absolute error (MAE), root-mean-square error (RMSE), and the probabilities of localizing within one and two samples of the true onset.

Because the evaluation data are sampled at 10 Hz, one sample corresponds to 0.1 s. Thus, the probabilities P_1 and P_2 represent localization within ±0.1 s and ±0.2 s, respectively. This temporal interpretation is important for navigation integrity: the objective is not merely to confirm that spoofing has occurred, but to identify how far back in the estimator history corrupted measurements may have begun influencing the state estimate [26].

4.2.2.1 Overall Onset Localization Performance

Table 4.13 summarizes the overall localization performance across all 18,034 held-out attacks [26]. CUSUM exhibits the largest errors, with a median error of 22 samples, MAE of 21.55 samples, and RMSE of 24.14 samples. C-GLR reduces the median error to 6 samples and the MAE to 11.62 samples. Both neural methods produce a median error of 0 samples, with the enhanced neural model achieving the lowest MAE and RMSE at 4.75 and 9.58 samples, respectively.

Table 4.13. GNSS spoofing onset-localization performance across 18,034 held-out realizations [26].

Method	Median Error (samples)	MAE (samples)	RMSE (samples)	P_1	P_2
CUSUM	22	21.55	24.14	0.016	0.026
C-GLR	6	11.62	17.11	0.241	0.281
NN-Parity	0	6.05	11.03	0.563	0.597
NN-Enhanced	0	4.75	9.58	0.605	0.658

At 10 Hz, the CUSUM median error corresponds to approximately 2.2 s and the C-GLR median error corresponds to approximately 0.6 s. The zero-sample median of both neural models means that at least half of their held-out onset estimates occur exactly at the true onset index. The enhanced neural model also achieves P_1 = 0.605 and P_2 = 0.658, meaning that 60.5% and 65.8% of the evaluated attacks are localized within ±0.1 s and ±0.2 s, respectively [26].

4.2.2.2 Onset Localization Error Distributions

Aggregate statistics do not fully describe the localization behavior because different methods can have similar mean errors while distributing those errors differently. The empirical cumulative distribution functions (CDFs) reported in the paper show a clear ordering of estimator precision [26]. Both neural curves rise sharply near zero error, whereas the CUSUM curve increases gradually over a much wider error range. C-GLR lies between these extremes.

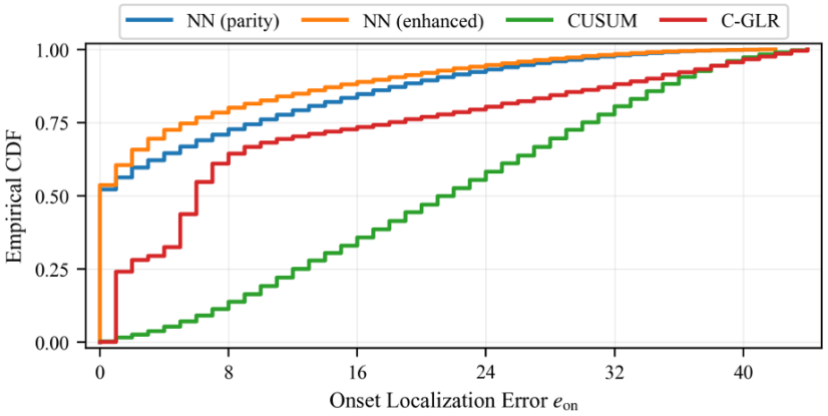


Figure 4.14. Empirical CDF of absolute GNSS spoofing onset-localization error for the four evaluated methods [26]. Neural localizers concentrate substantially more probability near zero error than CUSUM and C-GLR.

The enhanced neural curve remains above the parity-feature neural curve over most of the error range, showing that the addition of Kalman gain, Normalized Innovation Squared (NIS), and state-covariance information provides useful context beyond the normalized residual sequence alone. The CDF therefore supports the same conclusion as Table 4.13 while also showing that the improvement is not confined to a small number of outlier cases [26].

4.2.2.3 CUSUM versus C-GLR Performance

The comparison between CUSUM and C-GLR isolates the benefit of explicitly modeling the structured mean shift and temporal covariance of the residual window. CUSUM estimates the onset retrospectively from accumulated residual energy, whereas C-GLR evaluates candidate onset indices using attack-profile templates and the covariance-weighted likelihood formulation developed in Section 3.2.2 [26].

C-GLR reduces MAE from 21.55 to 11.62 samples, an improvement of approximately 46.1%, and reduces RMSE from 24.14 to 17.11 samples, an improvement of approximately 29.1%. The probability of estimating the onset within one sample increases from 0.016 for CUSUM to 0.241 for C-GLR, while P_2 increases from 0.026 to 0.281 [26]. These results confirm that retrospective localization benefits substantially from modeling the structure of the attack-induced residual change rather than relying only on sequential accumulation.

The improvement does not eliminate the limitations of a parametric localizer. C-GLR assumes step or ramp mean-shift templates, and the source paper notes that this rigidity limits performance when the observed residual evolution is not well represented by the assumed template. This is especially important for gradual attacks. More generally, evidence-accumulation procedures such as CUSUM inherently trade false-alarm control against response delay [31], [32], [34], which helps explain why an alarm time should not be treated as an exact estimate of attack onset.

4.2.2.4 Neural Onset Localization Performance

The two neural localizers provide the best overall performance. NN-Parity uses normalized residuals, residual energy, and first-order residual differences. NN-Enhanced adds internal UKF diagnostics, including the Kalman gain, NIS, and the trace of the state covariance. Both models

achieve a median error of 0 samples, demonstrating substantially tighter localization than either model-based baseline [26].

The enhanced feature set reduces MAE from 6.05 to 4.75 samples and RMSE from 11.03 to 9.58 samples relative to NN-Parity. Compared with C-GLR, the enhanced neural localizer reduces MAE by approximately 59.1% and RMSE by approximately 44.0%. Compared with CUSUM, the corresponding reductions are approximately 78.0% and 60.3%. The improvement in P_1 from 0.563 to 0.605 and in P_2 from 0.597 to 0.658 further indicates that estimator diagnostics improve the probability of very precise onset reconstruction [26].

These results support the use of data-driven temporal representations when the attack signature and estimator behavior interact in ways that are difficult to capture with a fixed parametric template. At the same time, the neural results are empirical and depend on the simulated training and test distributions. They therefore demonstrate superior performance within the evaluated KITTI-derived scenarios rather than a universal guarantee for arbitrary GNSS spoofing profiles.

4.2.2.5 Effect of Spoof Energy on Localization Accuracy

The covariance-weighted residual spoof energy, E, provides a direct link between the statistical detectability analysis and the observed localization performance. At low energy, the attack-induced mean shift is weak relative to measurement uncertainty, so neighboring onset hypotheses are difficult to distinguish. The paper reports elevated median localization errors for all methods in this regime [26], consistent with the detectability formulation introduced in Section 3.2.2.3 and standard Gaussian detection theory [33].

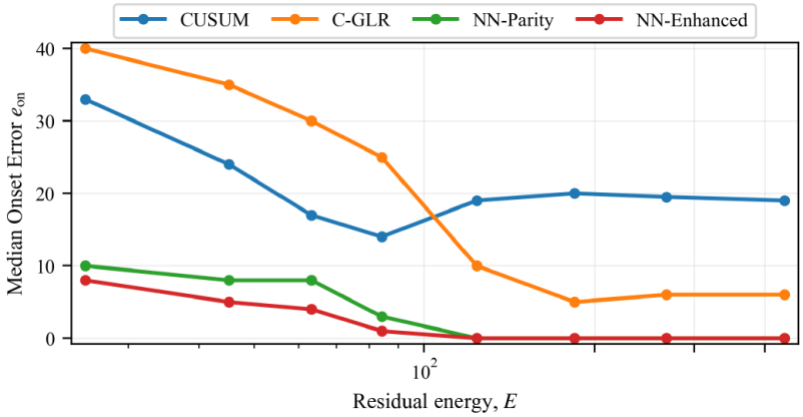


Figure 4.15. Median onset-localization error as a function of covariance-weighted residual spoof energy [26]. Greater spoof energy improves onset separability, with neural methods maintaining the lowest errors across the evaluated range.

As residual spoof energy increases, C-GLR exhibits a pronounced decrease in median error before stabilizing. The two neural estimators maintain lower error and eventually reach zero median error, while CUSUM improves only modestly before reaching a plateau [26]. This result shows that onset localization is fundamentally more uncertain when the attack signature is weak. It also confirms that the benefit of the more expressive localizers becomes most relevant when the available residual evidence is limited.

4.2.2.6 Step versus Ramp Spoofing Results

Attack temporal structure produces a second important source of localization difficulty. The evaluation contains 8,997 step-spoofing realizations and 9,037 ramp-spoofing realizations [26]. Step attacks introduce an immediate bias, creating a relatively distinct change in the residual sequence. Ramp attacks begin gradually, so the early post-onset residual signature can remain close to the nominal distribution and neighboring onset candidates become more difficult to separate.

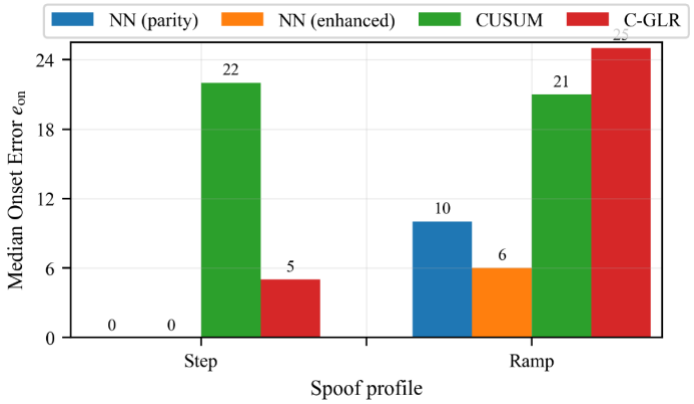


Figure 4.16. Median onset-localization error for step and ramp GNSS spoofing profiles [26]. Gradually evolving ramp attacks are more difficult to localize because their early residual signatures are weaker and less distinguishable.

For step attacks, C-GLR achieves low error and both neural models reach zero median error, while CUSUM remains less precise [26]. Ramp attacks increase localization error for most methods, with the most pronounced degradation occurring for C-GLR. This behavior is consistent with the theoretical analysis: the assumed ramp profile accumulates energy over time but begins with small residual deviations, which reduces local separability near the actual onset.

The neural localizers remain the most accurate under the ramp condition, suggesting that learned temporal features are better able to exploit subtle evolution in the residual sequence than the fixed parametric templates used by C-GLR. CUSUM changes relatively little between step and ramp cases, but its overall error remains substantially larger; this apparent insensitivity therefore reflects limited localization precision rather than superior handling of ramp attacks [26].

4.2.2.7 Overall Robustness and Limitations

Taken together, the onset-localization results establish three main findings. First, the alarm time produced by a sequential integrity monitor can differ substantially from the true attack onset, as illustrated by the 22-sample median error of the retrospective CUSUM baseline. Second, correlation-aware statistical localization substantially reduces this error, demonstrating the value of covariance and attack-profile information. Third, the neural localizers provide the most accurate and robust estimates across the evaluated attacks, with the enhanced model achieving a 0-sample median error, 4.75-sample MAE, and 9.58-sample RMSE [26].

The results also identify clear operating limitations. Localization becomes less reliable at low residual spoof energy, and gradually evolving ramp attacks are more difficult than step attacks

because their early signatures are weak. C-GLR is limited by fixed parametric templates, whereas the neural models are dependent on the representativeness of their training distribution. The current experiments use simulated GNSS biases on KITTI-derived 30 s trajectories processed through a configured UKF, rather than field-collected spoofing attacks on an operational vehicle [26], [29], [30].

Accordingly, the demonstrated result is a validated residual-domain framework for retrospective onset estimation under the evaluated simulation conditions. Integration of the estimated onset into estimator rollback, state recovery, or resilient navigation remains future work, as does validation under broader attack profiles, different navigation filters, and real GNSS spoofing experiments. The source manuscript explicitly identifies flexible attack representations and estimator recovery as future directions [26]. These distinctions are important for the final CARMEN+ report because they separate the demonstrated V&V contribution from capabilities that are anticipated but not yet experimentally established.

5. Implications for Transportation

Explain how the results advance transportation practice, technology, safety, or policy, and how they can be deployed or used by practitioners and stakeholders.

5.1 Overall Transportation Significance

Cybersecurity in Highly Automated Transportation Systems (HATS) is directly connected to transportation safety because automated vehicle functions depend on the integrity of sensing, localization, state estimation, decision making, and control. A cyberattack that corrupts these information flows can propagate beyond the information layer and influence vehicle routing, trajectory planning, or physical behavior. The project therefore treats cybersecurity not only as a communications or computing problem, but as a transportation assurance problem in which malicious information must be detected, its reliability must be assessed, and safe operation must be supported when a trusted information source becomes unavailable or compromised [4], [5], [13].

The completed research addresses this need through two complementary subtasks. Subtask 1 develops methods for detecting and identifying Global Navigation Satellite System (GNSS) spoofing attacks and extends the work toward navigation when GNSS can no longer be trusted [10]-[12], [23]. Subtask 2 addresses verification and validation (V&V) by estimating the expected reliability of data-driven anomaly-detection models and by determining when a deceptive GNSS attack actually began within the delay preceding an integrity-monitor alarm [25], [26]. Together, the subtasks address both the immediate recognition of cyber threats and the confidence needed to interpret cybersecurity decisions in safety-critical transportation systems.

A common feature across the project is the use of data-driven methods to address uncertainty and nonlinearity that are difficult to capture using fixed analytical assumptions alone. Vehicle motion, sensor errors, cyber-traffic characteristics, and nonlinear estimator residuals can all vary with

operating conditions. The developed approaches use learned motion relationships, adaptive statistical boundaries, temporal models, fuzzy similarity analysis, and residual-domain change-point estimation to characterize these behaviors from data. This provides a practical pathway for transportation systems in which assurance mechanisms must operate under changing environmental, vehicle, and attack conditions rather than under a single fixed nominal model.

From a transportation practice perspective, the work supports a layered view of resilience. Detection methods can provide an early indication that localization information is no longer trustworthy; V&V methods can establish whether the associated data-driven models provide a credible basis for decision making; onset localization can bound the interval during which a recursive state estimate may have been corrupted; and alternative onboard sensing can support future navigation continuity after GNSS compromise. This progression aligns with the CARMEN+ objective of reducing transportation cybersecurity risks and improving resilient automated mobility under degraded, spoofed, or otherwise compromised Positioning, Navigation, and Timing (PNT) conditions [13].

5.2 Subtask 1: Detection and Identification

Subtask 1 advances transportation cybersecurity by progressively addressing increasingly difficult GNSS spoofing conditions. The work begins with data-driven motion-consistency monitoring, extends to adaptive anomaly detection and temporal analysis of gradual attacks, and then investigates stereo visual navigation as an ongoing post-detection capability. The broader transportation implication is a shift from treating GNSS as an implicitly trusted positioning source toward continuously evaluating whether GNSS-reported motion remains consistent with independently observed vehicle behavior.

5.2.1 Improved Cyberattack Detection for Automated Transportation

The first practical contribution is improved sensitivity to both conventional and stealthy GNSS spoofing attacks. The ICAIC data-driven framework used speedometer and gyroscope information to predict short-horizon vehicle displacement independently of GNSS and compared that prediction with GNSS-derived motion [10]. For conventional turn-by-turn, stop, and overshoot attacks, the method preserved very high detection performance, while the learned displacement representation reduced the nominal displacement threshold from 3.35 m for the analytical position-reconstruction baseline to 1.79 m. This tighter error boundary is important for transportation safety because smaller malicious trajectory deviations can be identified before they accumulate into larger localization consequences.

The adaptive DBSCAN framework further reduces dependence on a fixed below-threshold decision boundary. By updating the DBSCAN neighborhood radius from the recursive mean and standard deviation of residuals accepted as non-anomalous, the detector can adapt to changes in nominal displacement-error behavior while excluding identified anomalies from the update process [11]. In the evaluated experiments, this mechanism detected multiple small biased attacks whose individual approximately 1.41 m steps remained below the conventional 1.79 m displacement threshold. For automated transportation, this provides a useful model for integrity monitoring in which legitimate sensor-error variability can evolve without requiring every operating condition to be represented by one static threshold.

The time-series framework addresses an even more difficult attack regime by exploiting recent residual history rather than relying only on instantaneous error magnitude [12]. The experiments demonstrated detection of gradual spoofing sequences with individual steps reduced to approximately 0.28 m, corresponding to about 1.5 standard deviations of the nominal displacement error. This result is relevant to HAV safety because a sophisticated attacker may intentionally keep individual manipulations close to normal sensor uncertainty while accumulating a large position error over time. Temporal evidence provides an additional mechanism for recognizing this behavior before the accumulated error becomes the only obvious indication of compromise.

For transportation practitioners, these results support a layered detector in which large structured attacks can be handled by direct displacement and speed consistency checks, changing nominal uncertainty can be addressed through adaptive clustering, and very small persistent deviations can be evaluated using temporal information. Such layering can improve situational awareness of localization integrity and provide downstream planning or control systems with an explicit indication that GNSS information should be treated with reduced confidence. The experiments were conducted using real-world driving data with controlled spoofing injection rather than on-road live spoofing, so field validation under additional receivers, environments, vehicle platforms, and attack implementations remains necessary before production deployment.

5.2.2 Practical Use of Existing Onboard Vehicle Sensors

A second transportation benefit is the use of sensing modalities that are already common in automated vehicles. The spoofing-detection frameworks use vehicle speed and inertial information as an independent description of physical motion and compare that information with GNSS-derived motion [10]-[12]. This architecture does not require the detection algorithm to depend exclusively on specialized GNSS anti-spoofing hardware or encrypted satellite signals. Instead, it exploits cross-sensor consistency among information sources that may already be available through the vehicle's Controller Area Network, speed sensors, and Inertial Measurement Unit.

This approach can reduce the integration barrier for future deployment because the cybersecurity function can be implemented primarily in software above existing sensing and data-acquisition layers. The learned displacement model operates on a short temporal window, the adaptive DBSCAN statistics are updated recursively with constant-memory calculations, and the time-series detector uses pre-trained models and recent residual histories [10]-[12]. These properties are favorable for embedded implementation because they avoid continuous full-trajectory optimization or computationally intensive GNSS signal processing.

The published detection studies describe their algorithms as suitable for real-time operation, but vehicle deployment still requires platform-specific timing, memory, processor-load, and fault-containment testing. In particular, a production implementation would need verified data synchronization, sensor-health monitoring, deterministic interfaces with the existing localization stack, and a defined response when the detector reports reduced GNSS integrity. The present results therefore support implementation feasibility and architectural compatibility, while hardware-in-the-loop and on-vehicle execution should be treated as the next deployment-validation stage rather than as already demonstrated production capability.

5.2.3 Resilient Localization and Navigation Under GNSS Compromise

Detection alone does not maintain vehicle localization after GNSS has been declared untrustworthy. For this reason, Subtask 1 progresses toward a post-detection navigation capability through the ongoing stereo visual navigation research [23]. The current framework estimates short-horizon planar global displacement directly from synchronized stereo imagery using Scale-Invariant Feature Transform correspondences, chained Kanade-Lucas-Tomasi tracking, compact geometric features, direction classification, and conditional XGBoost prediction. The objective is not to reproduce a complete mapping or simultaneous localization and mapping system, but to provide locally defined displacement increments that can support continued motion estimation when satellite positioning is compromised or denied.

The experimental results demonstrate that useful global displacement information can be extracted from a small number of stereo-temporal feature correspondences. The current manuscript reports test direction-classification accuracies of 88.50% and 88.00% for the two global displacement axes, three-standard-deviation prediction-error bounds of 19.54 mm and 19.86 mm, and trajectory reconstruction with an Absolute Trajectory Error root-mean-square error of 210.92 mm and a translational Relative Pose Error root-mean-square error of 7.12 mm [23]. These results establish the feasibility of the visual displacement formulation within the controlled experimental platform.

An important deployment property is that the VICON motion-capture system used in the experiments is required only for ground-truth generation and model evaluation; it is not an input to the visual predictor during inference [23]. This distinction supports the longer-term goal of transferring the method from the miniature-city validation environment to vehicle-mounted stereo sensing. The compact 101-feature representation and tree-based prediction architecture were also selected to reduce dependence on large neural perception models and expensive full-state optimization, which is advantageous for future resource-constrained edge implementation.

The visual-navigation contribution remains ongoing and should therefore be interpreted as a demonstrated research prototype rather than a deployment-ready GNSS replacement. End-to-end runtime at the 20 Hz camera acquisition rate has not yet been established in the current report, and the evaluation is based on an indoor miniature vehicle and controlled scene. Future work identified in the manuscript includes larger-scale outdoor validation, visual-inertial navigation, and filtering of features associated with dynamic objects [23]. These extensions are necessary before the approach can be considered a robust fallback navigation capability for full-scale HAV operation.

5.2.4 Transportation Safety and Deployment Implications

Collectively, the Subtask 1 results support a transportation architecture in which localization integrity is monitored continuously rather than only after an obvious navigation failure. Earlier recognition of spoofing-induced inconsistency can reduce the period during which corrupted GNSS information is allowed to influence route selection, state estimation, and control. The methods also provide different levels of sensitivity for different threat regimes, which is important because an operational integrity monitor must respond to both large abrupt attacks and smaller manipulations designed to resemble normal uncertainty.

A practical deployment could use the developed methods as one layer within a broader localization-integrity manager. The detector output would not directly replace existing vehicle safety logic; instead, it could modify the confidence assigned to GNSS, trigger corroboration using other sensing modalities, or initiate a transition toward an alternative navigation mode. This type of confidence-aware architecture is more appropriate for safety-critical use than treating a single anomaly decision as an automatic control command.

The research also has implications for testing and assurance practice. Transportation developers can evaluate detector performance across attack magnitude, temporal profile, sensor uncertainty, and vehicle-motion conditions rather than reporting only aggregate accuracy. Agencies, test laboratories, and system integrators could use such evaluations to define scenario-based cybersecurity test plans for PNT-dependent automated systems. The present project does not propose a specific regulatory threshold or certification requirement; rather, it provides experimentally grounded methods and metrics that can contribute to future assurance procedures for operation under unreliable or maliciously manipulated GNSS.

5.3 Subtask 2: Data-Driven Verification and Validation of Attacked HATS

Subtask 2 addresses the reliability of cybersecurity decisions themselves. A detection model may report high performance, but transportation deployment requires evidence that the underlying data support reliable separation between normal and malicious behavior and that the timing of a detected attack can be reconstructed accurately enough to support post-alarm reasoning. Frederick's two contributions address these complementary V&V needs through fuzzy model-performance validation and residual-domain GNSS spoofing onset localization [25], [26].

5.3.1 Improving Confidence in Data-Driven Cybersecurity Models

The fuzzy equivalence relation framework provides a pre-deployment method for evaluating the structure of an anomaly-detection dataset before treating a trained model's reported test accuracy as sufficient evidence of reliability [25]. Rather than introducing another Distributed Denial of Service (DDoS) classifier, the method measures intra-class cohesion and inter-class ambiguity through fuzzy tolerance and equivalence relations and varies the alpha-cut to determine how strongly benign and malicious samples can be separated.

For the CIC-DDoS2019 data used in the study, the analysis identified an alpha-cut of 95.5% at which Adjusted Accuracy reached 67.24% and the estimated peak accuracy was 89.57% [25]. The important transportation implication is not the numerical value itself, which is dataset-specific, but the validation concept: model performance can be interpreted relative to an independently estimated level of class distinguishability. This provides an additional check against accepting a high machine-learning accuracy value without examining whether the dataset contains substantial overlap, ambiguity, or false-alarm risk.

For safety-critical transportation cybersecurity, this approach could support model-development gates in which the dataset is evaluated before or alongside classifier training. If the available examples do not support adequate separation between benign and malicious behavior, additional data collection, feature engineering, or scenario coverage may be required before deployment. Conversely, when a model's achieved accuracy is consistent with a data-derived benchmark, the

result provides stronger evidence that performance is not merely an artifact of one training configuration.

The current demonstration is specific to DDoS traffic and CIC-DDoS2019, so the reported benchmark cannot be transferred directly to GNSS spoofing, vehicle-network intrusion, or other transportation cyber threats. However, the V&V principle is broader: data-driven safety and cybersecurity models should be evaluated not only by final predictive metrics, but also by the inherent separability and ambiguity of the data used to develop them. Applying the framework to other HATS datasets would require new similarity analysis and new validation thresholds for each application [25].

5.3.2 Verification of Cyberattack Timing and State-Estimation Integrity

The onset-localization contribution addresses a different assurance problem that begins after an integrity monitor has already raised an alarm. Sequential GNSS spoofing detectors accumulate evidence before declaring an attack, so the alarm time can occur after the true attack onset. During this delay, corrupted GNSS measurements can already propagate through a recursive nonlinear navigation filter and bias the estimated vehicle state [26]. Treating the alarm time as the beginning of corruption would therefore underestimate the portion of estimator history that may be untrustworthy.

The proposed framework retrospectively searches an alarm-conditioned residual window to estimate the true change point. The statistical formulation explicitly accounts for temporal correlation in nonlinear estimator residuals, and the study compares CUSUM, a correlated Generalized Likelihood Ratio localizer, and two neural localizers [26]. Across 18,034 held-out spoof realizations, the enhanced neural configuration achieved a median onset error of zero samples, a mean absolute error of 4.75 samples, and a root-mean-square error of 9.58 samples. At the 10 Hz evaluation rate, the method also achieved a 0.605 probability of placing the onset estimate within one sample of the true onset [26].

For transportation integrity monitoring, the value of this result is the ability to bound when the navigation solution may have started to become corrupted. An onset estimate can identify which recent state estimates, measurements, or derived decisions require additional scrutiny after the alarm. This is particularly relevant for automated vehicles because downstream planning and control functions can consume recursively estimated states that were influenced by spoofed measurements before the detector accumulated sufficient evidence to trigger.

The results also show that localization confidence depends on attack characteristics. Higher covariance-weighted spoof energy improves the statistical separation of onset hypotheses, while ramp attacks are more difficult to localize than abrupt step attacks because their early residual signatures are weaker [26]. This distinction is important for HATS assurance because the uncertainty associated with the estimated onset should be considered explicitly rather than assuming identical confidence for all attack profiles.

5.3.3 Implications for Recovery and Resilient Navigation

Onset localization provides information that can support future recovery, but recovery itself was not demonstrated in the present contribution. The onset paper identifies integration with estimator

recovery as future work [26]. Accordingly, the transportation implication should be framed as enabling rather than completed capability: once an attack start time is estimated, a navigation system can determine how far back in the estimator history potentially corrupted measurements and states extend.

This information could support several recovery strategies. A system could reinitialize from a state preceding the estimated onset, replay trusted measurements while excluding compromised GNSS information, or use an alternative sensor source to reconstruct the affected interval. The estimated onset could also be used to define the time span that must be examined during post-event analysis. These uses are especially relevant to recursive state estimators, where contamination can persist after the original spoofed measurement has been received.

The onset-localization framework therefore provides a conceptual bridge between attack detection and resilient navigation. Subtask 1 determines that GNSS information is inconsistent or malicious, while Subtask 2 can estimate when the estimator began to be affected. A recovery manager could use both pieces of information to decide which state history is trustworthy and when an alternative localization mode should take precedence. However, such a closed-loop recovery architecture requires additional experimental validation before it can support safety-critical vehicle control.

The neural localizers' comparatively strong performance on ramp attacks is promising for gradual spoofing, but the current results are based on simulated GNSS biases applied to KITTI-derived trajectories and a configured Unscented Kalman Filter [26], [29], [30]. Real receiver behavior, different navigation-filter structures, multipath, sensor faults, and field spoofing conditions can alter the residual statistics. Broader testing is therefore required before onset estimates can be assigned operational confidence in production transportation systems.

5.3.4 Broader V&V Implications for Transportation Cybersecurity

Altogether, the two V&V contributions emphasize that cybersecurity performance in HATS should be accompanied by quantitative evidence of confidence. The fuzzy framework examines whether a dataset can support the expected anomaly-detection performance, while the onset framework evaluates whether the timing of a detected attack can be reconstructed from estimator evidence [25], [26]. Both approaches move beyond a single binary attack label and characterize properties that are important for interpreting and acting on that label.

This perspective can be useful to transportation manufacturers, system integrators, test laboratories, and researchers developing machine-learning-based assurance functions. Model validation can be tied to dataset separability, false-alarm behavior, and expected performance ranges; post-alarm validation can be tied to onset-estimation error, attack energy, temporal profile, and estimator characteristics. These quantities provide a more transparent basis for deciding where a cybersecurity model performs reliably and where additional safeguards or fallback behavior are required.

The methods are not yet a complete certification framework and should not be interpreted as defining universal acceptance criteria. The fuzzy benchmark must be recalculated for each relevant dataset and threat model, and the onset localizer requires validation across additional navigation filters, attack representations, and real-world spoofing conditions. Their broader contribution is

methodological: they demonstrate how data structure and estimator residual evidence can be converted into explicit V&V information for safety-critical cyber-physical transportation systems.

5.4 Combined Impact on HATS Safety and Resilience

The two subtasks form a complementary sequence rather than independent research tracks. Subtask 1 provides mechanisms for recognizing when localization information has become inconsistent with trusted vehicle motion and investigates an alternative visual source for post-compromise displacement estimation. Subtask 2 evaluates the credibility of data-driven cybersecurity models and reconstructs the time at which an identified GNSS attack began. Together, they address the questions of whether an attack is occurring, how much confidence should be placed in the detection process, when estimator corruption likely started, and how localization continuity may eventually be maintained after GNSS is rejected [10]-[12], [23], [25], [26].

Attack Detection and Identification -> Reliability Assessment -> Attack-Onset Reconstruction -> Resilient Navigation and Recovery

This progression supports improved situational awareness before, during, and after a cybersecurity alarm. Before deployment, V&V can provide evidence that the data used by an anomaly-detection model support meaningful separation between benign and malicious conditions [25]. During vehicle operation, displacement-, speed-, adaptive-, and time-series-based integrity monitoring can identify GNSS behavior that is inconsistent with physical vehicle motion [10]-[12]. After an alarm, onset localization can identify the portion of recent estimator history that may already have been corrupted [26]. If GNSS is then removed from the trusted sensor set, stereo visual displacement prediction provides an ongoing research path toward maintaining local motion information without satellite positioning [23].

A future integrated HATS architecture could use these capabilities as successive layers of an assurance manager. The detection layer would produce an integrity warning and identify the affected localization source. The V&V layer would provide information about the expected reliability of the underlying cybersecurity models. The onset-localization layer would estimate the earliest potentially corrupted state. A navigation-resilience layer could then select trusted sensor information, reinitialize or recover the estimator as appropriate, and continue operation under a defined degraded mode. The present project establishes important components of this sequence, but full closed-loop integration remains a future research and validation objective.

For transportation stakeholders, the combined work encourages scenario-based cybersecurity evaluation rather than reliance on a single detector or accuracy value. Relevant test dimensions include abrupt and gradual spoofing, below-threshold attack magnitude, temporal persistence, sensor uncertainty, model false alarms, estimator residual correlation, attack-onset uncertainty, and the availability of alternative localization sources. Evaluating these dimensions together can help identify failure modes that may not be visible when detection, V&V, and recovery are assessed separately.

The overall project therefore contributes to the CARMEN+ and U.S. Department of Transportation objective of reducing transportation cybersecurity risks by strengthening multiple stages of the resilience process [13]. The demonstrated methods improve attack awareness, quantify the

reliability of data-driven cybersecurity evidence, and provide a foundation for identifying corrupted estimator history and transitioning toward GNSS-independent motion estimation. Continued field validation, edge implementation, estimator-recovery integration, and full-scale vehicle testing are required to translate these research results into deployable transportation technology, but the combined framework provides a coherent pathway toward safer and more cyber-resilient HATS.

6. Workforce Development

The project involved seven students, including two Ph.D. students, one master's student, and four undergraduate students. Their activities supported research and training in transportation cybersecurity, machine learning, automated vehicles, GNSS spoofing, anomaly detection, verification and validation, computer vision, and resilient navigation.

6.1 Graduate Student Involvement

The project involved two Ph.D. students and one master's student. Their research activities included GNSS spoofing detection, data-driven anomaly detection, verification and validation, attack-onset localization, machine learning, and visual navigation under GNSS-compromised and GNSS-denied conditions.

The graduate students also contributed to data processing, algorithm development, experimental evaluation, technical writing, and preparation of research publications and presentations.

6.2 Undergraduate Student Involvement

The project involved four undergraduate students who participated in research activities related to automated transportation, cybersecurity, data analysis, experimental support, and machine-learning applications.

Their participation provided early exposure to transportation cybersecurity research and helped develop practical experience with research methods, data processing, and intelligent transportation technologies.

6.3 Research Training and Technical Skills

Students developed experience in several technical areas relevant to automated transportation, including:

- cybersecurity and GNSS spoofing;
- machine learning and anomaly detection;
- automated vehicles and navigation;
- verification and validation;
- computer vision and sensor-data processing;
- experimental data collection and analysis; and
- scientific writing and publication.

6.4 Student Publications and Conference Participation

Graduate students contributed to peer-reviewed journal and conference publications and presentations, including work published or presented at the IEEE 4th International Conference on AI in Cybersecurity (ICAIC), IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE Open Journal of Vehicular Technology (OJVT), and IFSA Winter Conference on Automation, Robotics & Communications for Industry 4.0/5.0 (ARCI).These activities provided experience in scientific writing, preparation of technical results, peer review, conference presentation, and communication of transportation cybersecurity research.

6.5 Contribution to Transportation Cybersecurity Workforce Development

The project provided interdisciplinary training across cybersecurity, artificial intelligence, sensing, navigation, and V&V for automated transportation systems. By involving students at the undergraduate, master's, and Ph.D. levels, the project contributed to preparing future researchers and engineers for the development, testing, and deployment of safer and more cyber-resilient transportation systems.

7. Publications and Presentations

1. A. Mohammadi, V. Hemmati, R. Ahmari, F. Owusu-Ambrose, M. N. Mahmoud, and A. Homaifar, “Detection of Multiple Small Biased GPS Spoofing Attacks on Autonomous Vehicles,” 2025 IEEE 4th International Conference on AI in Cybersecurity (ICAIC), Houston, TX, USA, 2025, pp. 1–9. DOI: 10.1109/ICAIC63015.2025.10849307.
2. A. Mohammadi, V. Hemmati, R. Ahmari, F. Owusu-Ambrose, M. N. Mahmoud, and A. Homaifar, “GPS Spoofing Attack Detection on Autonomous Vehicles Using Modified DBSCAN with Dynamic Threshold,” Proceedings of the 5th IFSA Winter Conference on Automation, Robotics & Communications for Industry 4.0/5.0 (ARCI 2025), 2025, pp. 74–76.
3. A. Mohammadi et al., “Detection of Multiple Small Biased GPS Spoofing Attacks on Autonomous Vehicles Using Time Series Analysis,” IEEE Open Journal of Vehicular Technology, vol. 6, 2025. DOI: 10.1109/OJVT.2025.3559461.
4. A. Mohammadi, R. Ahmari, V. Hemmati, F. Owusu-Ambrose, M. N. Mahmoud, P. Kebria, A. Homaifar, and M. Saif, “GPS Spoofing Attack Detection in Autonomous Vehicles Using Adaptive DBSCAN,” 2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2025, pp. 6247–6252.
5. F. Owusu-Ambrose, A. R. Nuhu, B. Lartey, A. Mohammadi, and A. Homaifar, “Fuzzy Equivalence Relation Clustering Framework for Distributed Denial of Service Detection Accuracy Estimation,” Proceedings of the 5th IFSA Winter Conference on Automation, Robotics & Communications for Industry 4.0/5.0 (ARCI 2025), 2025, p. 125.
6. F. Owusu-Ambrose, A. R. Nuhu, A. Mohammadi, Y. Ayalew, V. Hemmati, P. Kebria, and A. Homaifar, “Onset Localization under Deceptive GNSS in Nonlinear State Estimators,” 2026 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2026, (accepted).
7. V. Hemmati, M. Behnia, A. Mohammadi, A. R. Nuhu, and A. Homaifar, “Mission-Based Quadcopter Flight Simulation,” 2024 AIAA DATC/IEEE 43rd Digital Avionics Systems

Conference (DASC), San Diego, CA, USA, 2024, pp. 1–7. DOI: 10.1109/DASC62030.2024.10749535.
8. Sarkar, M., Yan, X., Gebru, B., Nuhu, A. R., Gupta, K. D., Vamvoudakis, K. G., & Homaifar, A. M. Sarkar et al., "A Data-Driven Approach for Performance Evaluation of Autonomous eVTOLs," in IEEE Transactions on Aerospace and Electronic Systems, vol. 61, no. 2, pp. 3626-3641, April 2025, doi: 10.1109/TAES.2024.3493853.
9. R. Ahmari, V. Hemmati, A. Mohammadi, P. Kebria, M. N. Mahmoud, and A. Homaifar, "A Data-Driven Approach for UAV-UGV Integration," Proceedings of the Automation, Robotics & Communications for Industry 4.0/5.0 (ARCI 2025), 2025, p. 77.
10. R. Ahmari, V. Hemmati, A. Mohammadi, M. Mynuddin, P. Kebria, M. N. Mahmoud, and A. Homaifar, "Evaluating Trojan Attack Vulnerabilities in Autonomous Landing Systems for Urban Air Mobility," Proceedings of the Automation, Robotics & Communications for Industry 4.0/5.0 (ARCI 2025), 2025, p. 80.
11. R. Ahmari, A. Mohammadi, V. Hemmati, M. Mohammed, M. N. Mahmoud, P. Kebria, A. Homaifar, and M. Saif, "An Experimental Study of Trojan Vulnerabilities in UAV Autonomous Landing," 2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2025, pp. 6253–6258.
12. V. Hemmati, A. Mohammadi, A. R. Nuhu, R. Ahmari, P. Kebria, and A. Homaifar, "Volatility in Certainty (VC): A Metric for Detecting Adversarial Perturbations During Inference in Neural Network Classifiers," Interdisciplinary Conference on Electrics and Computer (INTCEC 2025), 2025.
13. R. Ahmari, A. Mohammadi, V. Hemmati, P. Kebria, and A. Homaifar, "Adaptive Dynamic Clustering for Streaming Data Using Genetic Algorithm," Interdisciplinary Conference on Electrics and Computer (INTCEC 2025), 2025.
14. V. Hemmati, Y. Ayalew, A. Mohammadi, R. Ahmari, P. Kebria, A. Homaifar, and M. Saif, "Conflict-Free Flight Scheduling Using Strategic Demand Capacity Balancing for Urban Air Mobility Operations," 2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2025, pp. 6259–6264.
15. L. Zeleke, B. Lartey, A. R. Nuhu, Y. Ayalew, P. Kebria, and A. Homaifar, "A Novel Flight Modeling Framework for Unmanned Aircraft in Realistic Airspace Encounters," 2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC), 2025, pp. 6273–6280.
16. L. Zeleke, B. Lartey, A. Homaifar, M. N. Mahmoud, P. Kebria, and E. Tunstel, "An Integrated Encounter Modeling and Simulation Tool for a Robust Detect-and-Avoid System Evaluation," Aerospace Science and Technology, 2025, 110613. DOI: 10.1016/j.ast.2025.110613.
17. Y. Ayalew, H. Z. Saghazadeh, P. Kebria, V. Hemmati, and A. Homaifar, "Risk-Informed Urban Air Mobility Flight Operations in Convective Weather Conditions and Aircraft Constraints," AIAA SCITECH 2026 Forum, 2026, Paper 0466. DOI: 10.2514/6.2026-0466.
18. J. M. Magalhaes, N. Tukenmez, K. Vamvoudakis, F. Fotiadis, A. Homaifar, and S. Bogosyan, "A Comprehensive Review of Trajectory Planning in Swarm Drones: Methods, Challenges, and Future Directions," AIAA SCITECH 2026 Forum, 2026, Paper 1769. DOI: 10.2514/6.2026-1769.
19. H. Z. Saghazadeh, Y. Ayalew, R. Ahmari, P. Kebria, and A. Homaifar, "Demand-Driven Vertiport Siting and Discrete-Event Fleet Simulation for On-Demand Urban Air Mobility Network Design," 2026 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2026, (accepted).

20. Girma, A. Homaifar, and M. N. Mahmoud, “ADP-Net: Adaptive Point Network with Multi-Scale Attention Mechanism for Small Object Detection,” Neurocomputing, 2025, 131381. DOI: 10.1016/j.neucom.2025.131381.
21. Y. Ayalew, P. Kebria, V. Hemmati, A. Homaifar, and E. Tunstel, “Risk Assessment and LLM Based Decision Support Framework for Urban Air Mobility Flight Operations in Hazardous Weather Conditions,” IEEE Transactions on Aerospace and Electronic Systems, 2025. DOI: 10.1109/TAES.2025.3645711.
22. R. Nuhu, P. Kebria, V. Hemmati, B. Lartey, M. N. Mahmoud, A. Homaifar, and E. Tunstel, “A Validation Strategy for Deep Learning Models: Evaluating and Enhancing Robustness,” IEEE Open Journal of the Computer Society, vol. 7, pp. 276–289, 2026. DOI: 10.1109/OJCS.2025.3650722.

8. Data Management and Dataset Statement

8.1 Public Datasets Used in the Project

The project used three external datasets: the Honda Research Institute Driving Dataset (HDD), CICDDoS2019, and the KITTI dataset. No other external dataset is claimed as a data source in this report. The original datasets remain governed by the access and license conditions established by their respective providers.

The Honda Research Institute Driving Dataset (HDD) was used for the GNSS spoofing-detection research in Subtask 1 [14]. HDD contains approximately 104 hours of naturalistic driving collected in the San Francisco Bay Area using an instrumented vehicle with multiple synchronized sensing modalities. The project used the vehicle-motion and localization information needed for the displacement-based, adaptive DBSCAN, and time-series GNSS spoofing studies. Honda Research Institute provides HDD for non-commercial research use to university-affiliated users through a request process. Dataset access is available at https://usa.honda-ri.com/hdd.

The Canadian Institute for Cybersecurity DDoS 2019 (CICDDoS2019) dataset was used for the fuzzy-equivalence V&V contribution in Subtask 2 [27]. CICDDoS2019 contains labeled benign and Distributed Denial of Service (DDoS) traffic and provides both raw network captures and extracted flow features. The project used the dataset to evaluate class similarity, ambiguity, and dataset-based performance benchmarks. The dataset is publicly available from the Canadian Institute for Cybersecurity at the University of New Brunswick. Its published license permits redistribution and reuse provided that the dataset and associated publication are cited. Dataset access is available at https://www.unb.ca/cic/datasets/ddos-2019.html.

The KITTI dataset was used for the GNSS spoofing onset-localization research in Subtask 2 [29]. The onset-localization experiments used KITTI-derived vehicle trajectories as the basis for generating simulated GNSS measurements and step and ramp spoofing realizations. KITTI provides synchronized real-world autonomous-driving sensor data and is available through the KITTI Vision Benchmark Suite. Dataset download requires user registration, and the dataset is provided under the Creative Commons Attribution-NonCommercial-ShareAlike 3.0 license. Dataset access is available at https://www.cvlibs.net/datasets/kitti/.

8.2 Project-Generated and Derived Research Data

The project generated derived research data from the three source datasets rather than creating an additional public benchmark dataset. For the HDD-based GNSS spoofing studies, project-derived artifacts include synthetically injected turn-by-turn, stop, overshoot, and multiple small biased attack realizations; processed sensor windows; displacement and speed residuals; and model training and evaluation files. For the CICDDoS2019 study, derived artifacts include the preprocessed feature sets, balanced analysis batches, fuzzy-similarity relations, and evaluation outputs used for V&V. For the KITTI-based onset-localization study, derived artifacts include simulated GNSS measurements, injected step and ramp spoofing realizations, nonlinear-estimator residual sequences, and model training and evaluation files. These project-derived artifacts are not represented in this report as independently published public datasets.

8.3 Visual Navigation Experimental Dataset

The ongoing visual-navigation research in Contribution 4 of Subtask 1 produced a project-specific experimental dataset. The data include synchronized stereo images, Inertial Measurement Unit (IMU) measurements where applicable, VICON motion-capture ground truth, synchronization information, and derived stereo-temporal feature datasets used for model development and evaluation.

The visual-navigation dataset is not publicly available at the time of this final report, and no public-access claim is made for the stereo images, VICON ground truth, synchronized sensor files, or derived feature datasets. The project team plans to prepare and publish the visual-navigation data within approximately two years, in coordination with completion and publication of Contribution 4. Until that release is completed, these data should be treated as non-public project research data.

8.4 Data Storage, Access, and Archiving

Access to the original HDD, CICDDoS2019, and KITTI data is governed by the respective source providers rather than by this project. The experimental visual-navigation data generated during this project are planned to be published after the ongoing Contribution 4 research is completed and the associated publication is released, with an anticipated release within approximately two years. The future dataset release is expected to include the stereo images, synchronized sensor measurements, ground truth, and documentation needed to facilitate research use and reproducibility.

9. Matching Funds Statement

9.1 Cost Matching Justification

In accordance with the University Transportation Centers (UTC) Grant Deliverables and Reporting Requirements, the following non-federal matching funds were applied to the project. For **Year 1,** the approved budgeted amount from UTC was **$300,000**, and the total non-federal match is **$138,147.79**. **Year 2,** cost-share periods for 2024–2025 Budgeted Amount from UTC was **$300,000** and the total non-federal match is **$91,747.62**, and **Year 3**, 2025–2026 non-federal match totals

$61,925.40, as described below. The cumulative non-federal match across all reported periods is **$291,820.81**. No federal funds were used to satisfy the non-federal match requirement.

The following are the detailed summary of matching funds for each year.

9.2 Year 1 Non-Federal Match for 2023–2024

9.2.1 Dr. Mahmoud

9.2.1.1 Cisco Grant Matching — $53,214

Dr. Mahmoud received a grant from Cisco for the proposal "Interpretable Machine Learning for High Stack Applications." Per Dr. Mahmoud's communication with Cisco representatives, Cisco agreed that these funds may be used as part of the cost match for this proposal.

9.2.1.2 ECE Departmental Matching — $12,233.19

Dr. Eroglu, Chair of the Department of Electrical and Computer Engineering, agreed to provide a 12% matching release for Dr. Mahmoud as support for this proposal during the academic year. Dr. Mahmoud's cost-share match covers the period 06/01/2023–05/31/2024 and totals $12,233.19. Fringe benefits and overhead are calculated at 24% and 44%, respectively.

9.2.2 Dr. Homaifar

9.2.2.1 Lockheed Martin Matching — $47,553.16

Dr. Homaifar contacted Lockheed Martin regarding the grant titled "Multi-Domain Autonomous Vehicle Control and Teaming." Per his communication with Lockheed Martin, the organization agreed that this fund may be used as part of the cost match for this proposal.

9.2.2.2 ECE Departmental In-Kind Matching — $25,147.44

Dr. Eroglu, Chair of the Department of Electrical and Computer Engineering, agreed to provide a 10% matching release for Dr. Homaifar as support for this proposal during the academic year. Dr. Homaifar's cost-share match covers the period 06/01/2023–05/31/2024 and totals $28,817.00. Fringe benefits and overhead are calculated at 24% and 44%, respectively.

9.2.3 Summary of year 1 Non-Federal Matching Funds

Cisco Grant — Dr. Mahmoud	$53,214.00
Lockheed Martin — Dr. Homaifar	$47,533.16
ECE Departmental Matching for Dr. Homaifar and Dr. Mahmoud	$37,380.63
Total	**$138,147.79**

9.3 Year 2 Non-Federal Match for 2024–2025

For the 2024–2025 period, additional non-federal cost-share matching was provided through departmental support. Dr. Homaifar contributed $69,676.74, consisting of salary of $39,585.05, fringe benefits of $8,801.58, and indirect costs of $21,290.11. Dr. Mahmoud contributed $22,070.88, consisting of salary of $11,900.44, fringe benefits of $3,426.56, and indirect costs of $6,743.88. The total additional matching amount for this period is $91,747.62.

9.4 Year 3 Non-Federal Match for 2025–2026

For the 2025–2026 project year, the NCAT University budget includes $61,925.40 in non-federal matching support. This consists of Dr. Homaifar's salary of $35,585.95, associated fringe benefits of $7,417.81, and indirect costs of $18,921.64. The overall project period runs from 06/01/2023 through 08/30/2026.

9.5 Cumulative Non-Federal Match Summary

Year 1 — Cisco, Lockheed Martin, and ECE Departmental Matching	$138,147.79
Year 2 — 2024–2025 Non-Federal Match	$91,747.62
Year 3 — 2025–2026 Non-Federal Match	$61,925.40
Cumulative Total	**$291,820.81**

9.6 Certification Statement

The non-federal matching funds listed above were applied to eligible project costs, including personnel, fringe benefits, tuition, travel, materials, publications, and indirect costs, as applicable. No federal funds were used to satisfy the non-federal match requirement.

10. Conclusions and Recommendations

10.1 Conclusions

This project developed and evaluated a coordinated set of data-driven methods for improving the cybersecurity, verification and validation (V&V), and resilience of Highly Automated Transportation Systems (HATS) under cyberattack. The findings show that transportation cybersecurity is strengthened when attack detection is treated as part of a broader assurance process that also evaluates model reliability, reconstructs the period of possible estimator corruption, and supports continued motion estimation after an external navigation source becomes untrustworthy.

The GNSS spoofing-detection research demonstrated a progressive improvement in sensitivity to increasingly stealthy attacks. The data-driven displacement framework reduced the nominal displacement-error threshold from 3.35 m for the analytical baseline to 1.79 m while maintaining strong performance for conventional turn-by-turn, stop, and overshoot attacks [10]. Adaptive

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) then extended detection below that fixed threshold and achieved 98.38% accuracy and 100% sensitivity for multiple small biased attacks with approximately 1.41 m individual displacement steps [11]. The time-series framework further incorporated recent displacement-error history and enabled real-time detection of smaller, more gradually evolving attacks. It achieved 95.92% accuracy for approximately 0.56 m steps over 25 s and 93.66% accuracy for approximately 0.28 m steps over 50 s [12]. The 0.28 m attack step corresponds to approximately 1.5 standard deviations of the nominal displacement error between GNSS-derived motion and motion estimated from trusted in-vehicle sensors. Collectively, these results demonstrate the value of combining direct motion consistency, adaptive statistical boundaries, and temporal evidence rather than relying on a single fixed threshold.

The project also extended the detection research toward navigation after GNSS compromise. The ongoing stereo visual-navigation framework demonstrated that short-horizon planar vehicle displacement can be estimated from independent onboard visual information using sparse stereo correspondences, temporal feature tracking, compact geometric features, direction classification, and Extreme Gradient Boosting models [23]. In the controlled indoor testbed, the two displacement-direction classifiers achieved 88.50% and 88.00% accuracy. The reconstructed trajectory achieved an Absolute Trajectory Error (ATE) root-mean-square error of 210.92 mm and a translational Relative Pose Error (RPE) root-mean-square error of 7.12 mm. These results support the feasibility of visual motion estimation after GNSS rejection, while also showing that accumulated trajectory drift remains and that the current method is a research prototype rather than a completed standalone replacement for GNSS.

The V&V research showed that the reliability of cybersecurity decisions should be assessed in addition to the final accuracy of a trained detector. The fuzzy-equivalence framework demonstrated a dataset-structure-based approach for estimating expected anomaly-detection performance by considering both within-class similarity and between-class ambiguity [25]. For the CIC-DDoS2019 data evaluated in the study, the selected 95.5% alpha-cut produced an Adjusted Accuracy of 67.24% and an estimated peak accuracy of 89.57%. These numerical values are dataset-specific and should not be interpreted as universal DDoS-detection limits; their significance is that they provide an independent benchmark against which a trained model's reported performance can be evaluated.

The GNSS spoofing onset-localization research addressed the complementary problem that an integrity-monitor alarm can occur after the true beginning of an attack. Across 18,034 held-out spoofing realizations, the enhanced neural localizer achieved a zero-sample median onset error, a 4.75-sample mean absolute error, and a 9.58-sample root-mean-square error [26]. The results also showed that low-energy attacks and gradually evolving ramp attacks are more difficult to localize than stronger or abrupt attacks. Onset localization therefore provides useful information for identifying which recent navigation states may already have been influenced before an alarm, although estimator rollback, state recovery, and closed-loop vehicle recovery were not demonstrated in the current work.

10.2 Recommendations and Future Research

Future research should first emphasize implementation and deployment-oriented validation. The developed methods should be evaluated on representative onboard and edge-computing hardware with measured average, worst-case, and end-to-end execution times. This is particularly important

for the stereo visual-navigation pipeline, for which the 20 Hz camera acquisition rate establishes a 50 ms throughput target but complete pipeline timing has not yet been demonstrated [23]. Testing should also be expanded across additional vehicle platforms, sensor configurations, road environments, operating speeds, weather and illumination conditions, and a broader range of attack magnitudes and temporal profiles.

The stereo visual-navigation research should be continued toward full-scale outdoor operation and publication of the ongoing VN study. Future development should examine integration with additional trusted onboard sensing, particularly inertial measurements, to reduce accumulated visual trajectory drift and improve robustness when visual information is temporarily weak. Dynamic-object filtering, changing illumination, camera vibration, longer feature distances, and complex traffic scenes should also be included in future evaluation before the method is considered as a practical fallback navigation capability.

The V&V framework should be evaluated on additional transportation cybersecurity datasets, attack classes, and feature representations to determine how well the fuzzy-equivalence methodology generalizes beyond the current binary DDoS case. The onset-localization framework should likewise be tested with additional navigation estimators, sensor combinations, detector-delay characteristics, attack profiles, and real or field-representative GNSS spoofing conditions. Particular attention should be given to quantifying confidence in the estimated onset for low-energy and ramp attacks, where the present results show greater localization difficulty.

A longer-term objective should be integration of the individual contributions into a coordinated resilience architecture. In such a system, attack detection would reduce confidence in compromised GNSS information, V&V methods would provide evidence regarding the reliability of the cybersecurity decision, onset localization would identify the interval of potentially corrupted state history, and trusted alternative sensing would support recovery or continued navigation. Future experiments should therefore investigate estimator rollback or reinitialization, replay of trusted measurements, transition logic between navigation modes, and closed-loop vehicle testing. This integrated validation is necessary before the developed methods can support deployment in safety-critical HATS.

References

[1] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, “A survey of autonomous driving: Common practices and emerging technologies,” IEEE Access, vol. 8, pp. 58443–58469, 2020.

[2] Y. Ma, Z. Wang, H. Yang, and L. Yang, “Artificial intelligence applications in the development of autonomous vehicles: A survey,” IEEE/CAA Journal of Automatica Sinica, vol. 7, no. 2, pp. 315–329, 2020.

[3] W. Zong, C. Zhang, Z. Wang, J. Zhu, and Q. Chen, “Architecture design and implementation of an autonomous vehicle,” IEEE Access, vol. 6, pp. 21956–21970, 2018.

[4] X. Sun, F. R. Yu, and P. Zhang, “A survey on cyber-security of connected and autonomous vehicles (CAVs),” IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 7, pp. 6240–6259, 2021.

[5] Z. Ju, H. Zhang, X. Li, X. Chen, J. Han, and M. Yang, “A survey on attack detection and resilience for connected and automated vehicles: From vehicle dynamics and control perspective,” IEEE Transactions on Intelligent Vehicles, vol. 7, no. 4, pp. 815–837, 2022.

[6] V. L. L. Thing and J. Wu, “Autonomous vehicle security: A taxonomy of attacks and defences,” in 2016 IEEE International Conference on Internet of Things (iThings), IEEE Green Computing and Communications (GreenCom), IEEE Cyber, Physical and Social Computing (CPSCom), and IEEE Smart Data (SmartData), pp. 164–170, 2016.

[7] N. Yadav, S. A. Ansar, and P. K. Chaurasia, “Review of attacks on connected and autonomous vehicles (CAV) and their existing solutions,” in 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), pp. 1–6, 2022.

[8] F. Owusu-Ambrose, A.-R. Nuhu, B. Lartey, A. Mohammadi, and A. Homaifar, “Fuzzy equivalence relation clustering framework for distributed denial of service detection accuracy estimation,” *Automation, Robotics & Communications for Industry 4.0/5.0*, vol. 4, no. 5.0, pp. 125–129, 2025.

[9] A.-R. Nuhu, K. D. Gupta, W. B. Bedada, M. Nabil, L. A. Zeleke, A. Homaifar, and E. Tunstel, “Negative selection approach to support formal verification and validation of blackbox models’ input constraints,” in *2022 IEEE Symposium Series on Computational Intelligence (SSCI)*, pp. 413–420, 2022.

[10] A. Mohammadi, V. Hemmati, R. Ahmari, F. Owusu-Ambrose, M. N. Mahmoud, and A. Homaifar, “Detection of Multiple Small Biased GPS Spoofing Attacks on Autonomous Vehicles,” in 2025 IEEE 4th International Conference on AI in Cybersecurity (ICAIC), pp. 1–9, 2025, doi: 10.1109/ICAIC63015.2025.10849307.

[11] A. Mohammadi, R. Ahmari, V. Hemmati, F. Owusu-Ambrose, M. N. Mahmoud, P. Kebria, A. Homaifar, and M. Saif, “GPS Spoofing Attack Detection in Autonomous Vehicles Using Adaptive DBSCAN,” in 2025 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 6247–6252, 2025, doi: 10.1109/SMC58881.2025.11343401.

[12] A. Mohammadi, R. Ahmari, V. Hemmati, F. Owusu-Ambrose, M. N. Mahmoud, P. Kebria, and A. Homaifar, “Detection of Multiple Small Biased GPS Spoofing Attacks on Autonomous Vehicles Using Time Series Analysis,” IEEE Open Journal of Vehicular Technology, vol. 6, pp. 1152–1163, 2025, doi: 10.1109/OJVT.2025.3559461.

[13] CARMEN+, University Transportation Center Proposal: Reducing Transportation Cybersecurity Risks, U.S. Department of Transportation University Transportation Centers Program.

[14] V. Ramanishka, Y.-T. Chen, T. Misu, and K. Saenko, “Toward driving scene understanding: A dataset for learning driver behavior and causal reasoning,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7699–7707, 2018, doi: 10.1109/CVPR.2018.00803. [Online]. Available: https://usa.honda-ri.com/hdd.

[15] C. C. Robusto, “The cosine-haversine formula,” The American Mathematical Monthly, vol. 64, no. 1, pp. 38–40, 1957.

[16] L. R. Agostinho, N. M. Ricardo, M. I. Pereira, A. Hiolle, and A. M. Pinto, “A practical survey on visual odometry for autonomous driving in challenging scenarios and conditions,” IEEE Access, vol. 10, pp. 72182–72205, 2022.

[17] J. Cheng, L. Zhang, Q. Chen, X. Hu, and J. Cai, “A review of visual SLAM methods for autonomous driving vehicles,” Engineering Applications of Artificial Intelligence, vol. 114, Art. no. 104992, 2022.

[18] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” International Journal of Computer Vision, vol. 60, no. 2, pp. 91–110, 2004.

[19] B. D. Lucas and T. Kanade, “An iterative image registration technique with an application to stereo vision,” in Proceedings of the 7th International Joint Conference on Artificial Intelligence (IJCAI), vol. 2, pp. 674–679, 1981.

[20] C. Tomasi and T. Kanade, “Shape and motion from image streams: A factorization method—Part 3: Detection and tracking of point features,” Carnegie Mellon University, Pittsburgh, PA, Tech. Rep. CMU-CS-91-132, Apr. 1991.

[21] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794, 2016.

[22] M. A. Fischler and R. C. Bolles, “Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography,” Communications of the ACM, vol. 24, no. 6, pp. 381–395, 1981.

[23] A. Mohammadi et al., “Stereo Visual Navigation for Global Displacement Prediction of Autonomous Vehicles in GNSS-Compromised Environments,” manuscript in preparation, 2026.

[24] W. Kabsch, “A solution for the best rotation to relate two sets of vectors,” Acta Crystallographica Section A, vol. 32, no. 5, pp. 922–923, 1976.

[25] F. Owusu-Ambrose, A.-R. Nuhu, B. Lartey, A. Mohammadi, and A. Homaifar, “Fuzzy Equivalence Relation Clustering Framework for Distributed Denial of Service Detection Accuracy Estimation,” in

5th IFSA Winter Conference on Automation, Robotics & Communications for Industry 4.0/5.0 (ARCI 2025), Granada, Spain, Feb. 19–21, 2025, pp. 125–131.

[26] F. Owusu-Ambrose, A.-R. Nuhu, A. Mohammadi, Y. Ayalew, V. Hemmati, P. Kebria, and A. Homaifar, "Onset Localization under Deceptive GNSS in Nonlinear State Estimators," manuscript, 2026.

[27] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, "Developing realistic Distributed Denial of Service (DDoS) attack dataset and taxonomy," in 2019 International Carnahan Conference on Security Technology (ICCST), pp. 1–8, 2019, doi: 10.1109/CCST.2019.8888419. Dataset: CICDDoS2019. [Online]. Available: https://www.unb.ca/cic/datasets/ddos-2019.html.

[28] T. J. Ross, "Fuzzy tolerance and equivalence relations," in Fuzzy Logic with Engineering Applications, 3rd ed. Hoboken, NJ, USA: John Wiley & Sons, 2010, pp. 48–66.

[29] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," International Journal of Robotics Research, vol. 32, no. 11, pp. 1231–1237, 2013, doi: 10.1177/0278364913491297. [Online]. Available: https://www.cvlibs.net/datasets/kitti/.

[30] S. J. Julier and J. K. Uhlmann, "A new extension of the Kalman filter to nonlinear systems," in Signal Processing, Sensor Fusion, and Target Recognition VI, vol. 3068, 1997, pp. 182–193.

[31] E. S. Page, "Continuous inspection schemes," Biometrika, vol. 41, no. 1/2, pp. 100–115, 1954.

[32] M. Basseville and I. V. Nikiforov, Detection of Abrupt Changes: Theory and Application. Englewood Cliffs, NJ, USA: Prentice Hall, 1993.

[33] S. M. Kay, Fundamentals of Statistical Signal Processing, Volume II: Detection Theory. Upper Saddle River, NJ, USA: Prentice Hall, 1998.

[34] H. V. Poor and O. Hadjiliadis, Quickest Detection. Cambridge, U.K.: Cambridge University Press, 2008.

[35] T. E. Humphreys, B. M. Ledvina, M. L. Psiaki, B. W. O'Hanlon, and P. M. Kintner, "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), 2008, pp. 2314–2325.

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

[37] N. O. Tippenhauer, C. Pöpper, K. B. Rasmussen, and S. Čapkun, "On the requirements for successful GPS spoofing attacks," in Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS), 2011, pp. 75–86.

[38] S. Behal and K. Kumar, "Trends in validation of DDoS research," Procedia Computer Science, vol. 85, pp. 7–15, 2016.

[39] M. R. Kadri, A. Abdelli, and L. Mokdad, "Survey and classification of DoS and DDoS attack detection and validation approaches for IoT environments," Internet of Things, p. 101021, 2023.

[40] R. Doriguzzi-Corin, S. Millar, S. Scott-Hayward, J. Martínez-del-Rincón, and D. Siracusa, “LUCID: A practical, lightweight deep learning solution for DDoS attack detection,” IEEE Transactions on Network and Service Management, vol. 17, no. 2, pp. 876–889, June 2020.

[41] V. Nguyen, T. Nguyen, N. Vu, and T. Hoang, “A new anomaly traffic detection based on fuzzy logic approach in wireless sensor networks,” in Proceedings of the 10th International Symposium on Information and Communication Technology, Hanoi, Vietnam, pp. 1–6, Dec. 2019.

[42] S. Huang, Y. Guo, N. Yang, S. Zha, D. Liu, and W. Fang, “A weighted fuzzy C-means clustering method with density peak for anomaly detection in IoT-enabled manufacturing process,” Journal of Intelligent Manufacturing, vol. 32, pp. 1845–1861, 2021.

[43] A. Alsirhani, S. Sampalli, and P. Bodorik, “DDoS detection system: Using a set of classification algorithms controlled by fuzzy logic system in Apache Spark,” IEEE Transactions on Network and Service Management, vol. 16, no. 3, pp. 936–949, 2019.

[44] D. Srilatha and G. K. Shyam, “Cloud-based intrusion detection using kernel fuzzy clustering and optimal type-2 fuzzy neural network,” Cluster Computing, vol. 24, pp. 2657–2672, 2021.

[45] V. M. Rios, P. R. M. Inácio, D. Magoni, and M. M. Freire, “Detection of reduction-of-quality DDoS attacks using fuzzy logic and machine learning algorithms,” Computer Networks, vol. 186, pp. 1–17, Jan. 2021.

[46] Y. Wang, J. Wang, X. Fan, and Y. Song, “Network traffic anomaly detection algorithm based on intuitionistic fuzzy time series graph mining,” IEEE Access, vol. 8, pp. 63381–63394, 2020.

[47] T. L. Lai, “Sequential changepoint detection in quality control and dynamical systems,” Journal of the Royal Statistical Society: Series B (Methodological), vol. 57, no. 4, pp. 613–644, 1995.

[48] A. Tartakovsky, I. Nikiforov, and M. Basseville, Sequential Analysis: Hypothesis Testing and Change-Point Detection. CRC Press, 2014.

[49] Y. Bar-Shalom, X.-R. Li, and T. Kirubarajan, Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software. John Wiley & Sons, 2001.

[50] B. D. O. Anderson and J. B. Moore, Optimal Filtering. Prentice-Hall, 1979.

[51] J. Liu, G. Guo, and R. Zhang, “Residual-based fault detection and exclusion with enhanced localization integrity,” IEEE Transactions on Vehicular Technology, vol. 72, no. 5, pp. 5798–5808, 2022.

[52] S. Jeong, M. Kim, and J. Lee, “CUSUM-based GNSS spoofing detection method for users of GNSS augmentation system,” International Journal of Aeronautical and Space Sciences, vol. 21, no. 2, pp. 513–523, June 2020.