
Awais Munir

# Machine Learning Based Strategy to Detect Meaconing attacks in GNSS

Method And Analysis Using FGI Dataset

School of Technology and Innovations
Master's thesis
Sustainable Autonomous System

Vaasa 2026

**UNIVERSITY OF VAASA**
**School of Technology and Innovation**
**Author:** Awais Munir
**Title of the Thesis:** Machine Learning Based Strategy to Detect Meaconing attacks in GNSS
**Degree:** Master of Science in Computing Sciences
**Programme:** Sustainable and Autonomous Systems
**Supervisor:** Mahmoud Elsanhoury
**Instructor:** Elham Ahmadi
**Year:** 2026 **Sivumäärä:** 83

**ABSTRACT:**
Global navigation satellite systems (GNSS) signals, which are used to guide modern navigation systems, can be subtly manipulated while preserving their authenticity. This thesis explores the use of machine learning (ML) for detecting meaconing attacks in GNSS to overcome fundamental vulnerabilities in navigation systems dependent on precise positioning. After conducting a review of GNSS signal processing, spoofing techniques and current methods of detection, this research examines the behaviour of the receiver level tracking loops during meaconing. The study focuses on the carrier-to-noise density ratio ($C/N_o$), Doppler frequency, delay-lock loop (DLL) discriminator, code phase, multi-correlator distortions. The study uses raw in-phase and quadrature (I/Q) tracking information collected from Finnish geospatial institute (FGI) and processed with the FGI-GSRx software defined receiver on MATLAB, to detect different patterns in these features during meaconing attacks. Sliding window segmentation is applied to capture temporal dynamics. ML models, including random forest (RF) and support vector machines (SVM) were trained to distinguish between authentic and attacked signals. The proposed framework shows high detection performance across GPS and Galileo constellations, with results indicating strong accuracy, low false alarm rates, and consistent performance under blocked chronological validation While individual features show varying sensitivity to meaconing, combining these features provides a strong discriminatory capability. Additionally, several analysis including feature correlation, dimensionality assessment, and satellite generalization were performed and confirmed robustness of the approach. The results suggest that tracking loop features combined with ML, offer a reliable and scalable solution for real time detection of meaconing attacks in GNSS receivers.

**KEYWORDS:** GNSS Security, Meaconing Detection, Spoofing, Machine Learning, Tracking-Loop Telemetry, Satellite-Independent Detection, Random Forest Algorithm, Support Vector Machine Algorithm

## Contents

## Figures



## Tables

## Abbreviations

| | |
|---|---|
| ADS-B | Automatic Dependent Surveillance-Broadcast |
| AGC | Automatic Gain Control |
| AJAMS | Anti Jamming and Meaconing System |
| AUC | Area Under the Curve |
| C/N0 | Carrier-to-Noise Density Ratio |
| CoMP | Coordinated Multi-Point |
| DD | Distance-Decreasing |
| DLL | Delay-Lock Loop |
| F1 | F1 Score |
| FAR | False Alarm Rate |
| FGI | Finnish Geospatial Institute |
| FGI-GSRx | Finnish Geospatial Institute GNSS Software Receiver |
| GNSS | Global Navigation Satellite System |
| GoF | Goodness-of-Fit |
| GPS | Global Positioning System |
| I/Q | In-phase and Quadrature |
| LOPO | Leave-One-PRN-Out |
| MDR | Missed Detection Rate |
| MIMO | Multiple-Input Multiple-Output |
| ML | Machine Learning |
| NWPR | Narrowband-Wideband Power Ratio |
| OSNMA | Open Service Navigation Message Authentication |
| PCA | Principal Component Analysis |
| PLL | Phase-Lock Loop |
| PMU | Phasor Measurement Unit |
| PNT | Positioning, Navigation, and Timing |
| PRN | Pseudo-Random Noise |
| RBF | Radial Basis Function |
| RF | Random Forest |
| ROC | Receiver Operating Characteristic |
| SCADA | Supervisory Control and Data Acquisition |
| SDR | Software-Defined Radio |
| SoL | Safety-of-Life |

| | |
|---|---|
| SVM | Support Vector Machine |
| TOA | Time of Arrival |
| TTD | Time-to-Detection |
| UAV | Unmanned Aerial Vehicle |
| URLLC | Ultra-Reliable Low-Latency Communication |

# 1 Introduction

This thesis investigates the detection of meaconing attacks in GNSS using tracking loop features extracting from software defined receiver data. The study includes signal processing analysis with machine learning (ML) techniques to identify anomalies in satellite tracking behaviour and evaluate the effectiveness of ML based detection methods for improving the security and reliability of GNSS dependent systems.

## 1.1 Context and Motivation

Modern cyber-physical systems depend on positioning, navigation, and timing (PNT) services provided by global navigation satellite systems (GNSS). While GNSS is commonly associated with navigation, its strategic value extends significantly beyond positioning. GNSS timing is widely used as the primary reference for synchronizing distributed infrastructure, including telecommunications, energy monitoring and protection systems, financial transaction timing, logistics coordination, and autonomous mobility platforms (Siemuri et al., 2021). In these settings, GNSS functions as a timing foundation for the digital systems: even small timing perturbations can propagate into operational instability and safety risks.

The dependency is becoming more consequential, with critical infrastructure becoming more interconnected and time sensitive. Base station synchronization in next-generation cellular networks is often based on GNSS timing to enable coordinated multi-point transmission (CoMP), ultra-reliable low-latency communication (URLLC), and advanced beamforming. Such timing errors at a microsecond level undermine coordination and quality of service, especially in the automation of industries and smart transport setups (H. Zhang et al., 2020). Similarly, power systems rely on GNSS timing to timestamp synchronized phasor measurements obtained from phasor measurement units (PMUs); compromised timing can alter phase-angle estimation and corrupt operator situational awareness (Wei & Sikdar, 2019). In transport systems with safety of life services, such as unmanned aerial vehicles (UAVs) and maritime platforms, subtle timing or position biases

would cause dangerous decisions in case external crosschecking infrastructure is not available (Manesh et al., 2019; Shafique et al., 2021).

Simultaneously with increased dependency, real-world GNSS interference has increased in operational settings. Recent analyses have reported a drastic rise in disruption of maritime navigation and timing-dependent operations, which supports the notion that GNSS disruption is no longer limited to laboratory experiments (Safi, 2025). This tendency increases the necessity of detection mechanisms that can be used in the presence of realistic receiver and environmental variability.

## 1.2 The Threat: Meaconing and why it is hard to detect

Meaconing attacks are particularly difficult to detect among GNSS threats due to their focus on preserving plausible receiver operation instead of denying service. It is a replay-based deception attack in which authentic GNSS signals are rebroadcasted by attacker with some delay. In contrast to simulator-generated spoofing, meaconing can leave navigation data content, modulation properties and possibly authentication related structures that may be transmitted within the signal intact (Marnach et al., 2013; Seco-Granados et al., 2021). This causes meaconing to be particularly harmful to timing dependent systems: a common replay delay across satellites can be presented primarily as a receiver clock bias, resulting in a stable but incorrect time reference rather than an inconsistent navigation solution.

Conventional detection mechanisms such as threshold monitoring of $C/N_0$, automatic gain control (AGC) excursions, or loss-of-lock, can be useful with high power jamming or naive spoofing, but are frequently ineffective against sophisticated replay attacks that attempt to be statistically plausible (Chen et al., 2022). Meaconing would not cause observable power anomalies and could track long enough to allow dependent systems to propagate errors in timing or position. With the increasing accessibility of software-defined radio (SDR) tooling and open GNSS processing stacks, replay-based experimentation can be more easily achieved, reducing operational barriers to attacks of operational relevance. (Islam, Bhuiyan, et al., 2024).

## 1.3 Problem Statement: Cross-Satellite Variability and Evaluation Leakage

GNSS receivers reveal extensive internal telemetry on the tracking-loop level. This telemetry has multivariate dynamics, which can capture both the initiation and development of the attack despite the absolute signal levels being realistic (Semanjski et al., 2020). ML is desirable because it has the ability to learn complex relationships between tracking features, that alone are weak indicators but together provide valuable predictive information.

But there is a significant methodological issue that constrains the operational reliability: distributions of tracking-loop features vary naturally among satellites pseudo-random noise (PRNs) and through time even in clean conditions. The causes of this variability are satellite elevation and geometry, antenna gain patterns and masking, multipath vulnerability, relative motion and doppler statistics and receiver channel characteristics. This can cause a classifier trained on absolute feature magnitudes to unintentionally learn satellite identity behaviour instead of attack signatures. This produces inflated performance under random window-based train/test splits including the same PRNs in training and testing and failing when tested on satellites not observed in training. Practically, the detector might be successful in validation but fail in deployment as the visible satellite set varies.

Accordingly, the central research question is:

How can meaconing be detected using tracking-loop features in a manner that generalizes across satellites without relying on satellite-specific absolute baselines?

## 1.4 Aim and Objectives

The main aim of this thesis is to design and evaluate an ML-based meaconing detection framework that is accurate, timely, and satellite-independent. The objectives are:

1. Extract compact, informative statistical features from GNSS tracking-loop outputs.

2. Estimate attack onset using a data-driven detection method to support reliable labeling.
3. Benchmark baseline supervised models (e.g., SVM, RF) for meaconing detection.
4. Evaluate performance under strict cross-satellite validation to test real generalization.
5. Design and test satellite-independent feature normalization using local per-satellite standardization.
6. Assess detection delay and robustness under different normalization assumptions.

## 1.5 Research Contributions

This thesis is aimed at receiver-level GNSS meaconing detection based on tracking-loop features of both GPS L1 and Galileo E1 signals. The main contribution is not simply the application of machine learning, but the method of assessing the detection problem under the conditions of satellite-independency. The first contribution is that it uses a small number of physically meaningful features derived out of receiver tracking outputs. These features describe $C/N_o$ behaviour, tracking-loop instability, code and Doppler variation, and correlator-shape distortion. The features are directly connected to the signal tracking process since the features are based on receiver-internal telemetry. The second contribution is the use of PRN-relative baseline normalization. Rather than applying the same absolute thresholds to all the satellites, each PRN is matched with its own clean pre-attack baseline. This minimizes the impact of satellite specific variations like signal strength, geometry, multipath and nominal tracking noise. The third contribution is that there is the leakage-conscious validation. The detector is tested with blocked chronological validation, multiple PRN split validation, PRN-disjoint testing, and LOPO validation. These tests are implemented to test whether the model is able to identify meaconing patterns on the hidden satellite groups and not just perform well when the same PRNs are present in both training and testing.

The thesis thus fills a practically important gap in ML-based GNSS detection: it is not enough to achieve high accuracy when random splits are used, unless the model is

learning PRN-based baselines. The work demonstrates that PRN-relative characteristics and more stringent validation offer a more credible evaluation of satellite-independent meaconing detection using the evaluated FGI-SpoofRepo DFMC dataset.

## 1.6 Thesis Structure

Chapter 2 reviews GNSS deception attacks and ML-based detection, emphasizing the cross-satellite variance problem and evaluation methodology. Chapter 3 presents the research methodology and the proposed detection pipeline, including signal modelling assumptions, feature extraction, local standardization, label, and strict cross-satellite validation. Chapter 4 presents the results followed by discussion on results interpretation in Chapter 5. At last, Chapter 6 concludes the study, states limitations of current study and presents future work possibilities. The overall organization of the thesis is illustrated in Figure 1, which shows how each chapter builds on the previous chapter and contributes to the proposed framework.

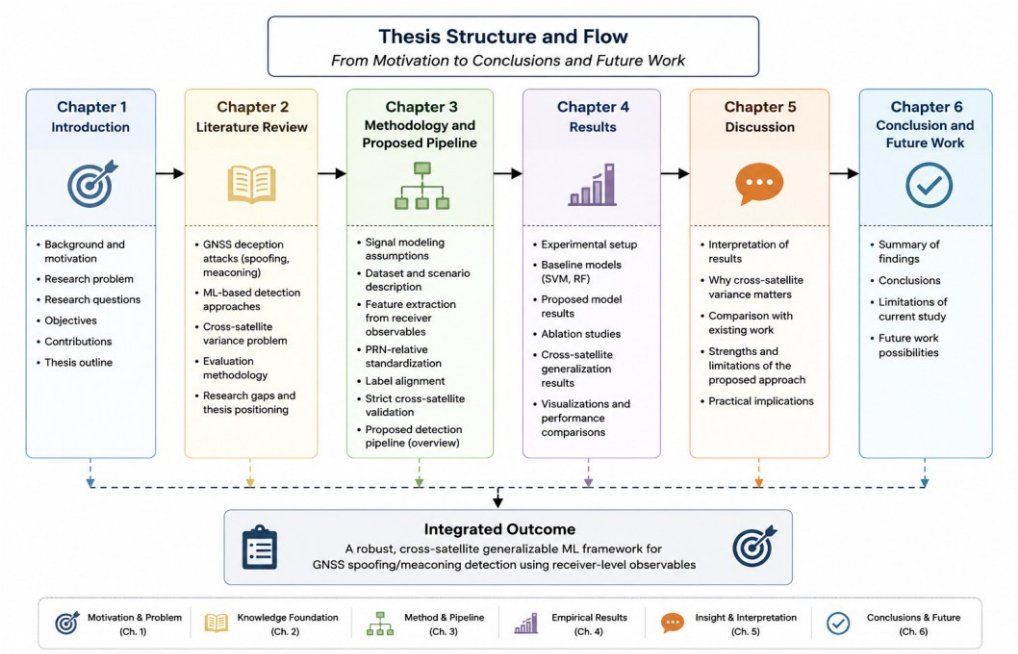


**Figure 1** Thesis structure overview.

# 2 Literature Review: Foundations of Adversarial GNSS Defense

The chapter includes a very extensive literature review of the GNSS security topic, especially the development of traditional threshold-based defense to ML-based detection models, and the critical problem of heterogeneous distribution of features among satellites. The review has been effectively structured into three fundamental themes including the receiver-level causes of inter-satellite variability, a realistic taxonomy of deception attacks and in particular meaconing and the evolution of machine learning paradigms towards robust and generalizable detection. The general objective of this review is not limited to the summarization of prior studies; it should help shed light on the methodological gap that this thesis aims to fill. Many published ML results are based on the conditions of data and protocols of validation that do not strictly test the cross-satellite generalization. This discontinuity is, in its turn, operational to the receivers, which leads to the current variability of the visible satellite set and the impossibility of the detector systems to use PRN-dependent baselines.

## 2.1 GNSS as Critical Infrastructure Timing Substrate

GNSS is often associated with positioning and navigation, but its role in modern cyber-physical systems also extends to timing and synchronization. Telecommunications networks, power grids, financial systems, logistics, distributed sensing, autonomous vehicles, UAVs, and maritime systems all depend on GNSS-derived timing or positioning information to coordinate distributed operations (Siemuri et al., 2021). Therefore, GNSS interference is not only a navigation problem. In many applications, it can also become a timing and system-integrity problem.

The effect of GNSS disruption has been reported across several domains. In telecommunications, timing errors can disturb synchronization between network components and degrade coordination in 5G and future communication systems. Timing errors on the order of microseconds may already affect coordination, while larger errors can disrupt

synchronization chains and reduce quality of service (H. Zhang et al., 2020). Empirical studies on 5G Stand Alone networks have also shown that jamming of GPS and GLONASS signals can push network management systems into a poor operating state and eventually cause loss of connection (Wührl et al., 2023). In power systems, GNSS timing is used by phasor Measurement Units to time-stamp voltage and current phasors. If GNSS timing is manipulated, phase-angle measurements and wide-area monitoring may be distorted, which can mislead state estimation and event localization (Wei & Sikdar, 2019; H. Zhang et al., 2020). Related work has shown that combining PMU and SCADA measurements with dynamic filtering can help estimate and correct phase-angle shifts caused by spoofing attacks (Siamak et al., 2021).

GNSS is also important in safety-of-life and autonomous systems. UAVs, autonomous vehicles, and maritime platforms use GNSS as part of broader sensor-fusion and motion-estimation chains (Shafique et al., 2021). In such systems, replay or spoofing attacks may introduce biased measurements while still maintaining a plausible signal appearance, that is particularly challenging when external cross-checks are limited or unavailable (Manesh et al., 2019). Prior studies have explored cooperative localization and trajectory-verification methods for UAVs and connected platforms as additional protection layers (Eldosouky et al., 2020;Dang et al., 2022). Recent reports also show that GNSS disruption has become increasingly relevant in real-world maritime and timing-dependent applications (Safi, 2025). These examples motivate the need for reliable GNSS interference and deception detection. However, the purpose of this thesis is not to provide a broad system-level review of every GNSS-dependent infrastructure. Instead, these application areas are used to establish why early receiver-level detection matters. The focus of this thesis is therefore narrowed to tracking-loop telemetry and machine-learning based meaconing detection, where the key challenge is not only detecting an attack, but evaluating whether the detector can generalize across different satellites and PRN-dependent signal conditions.

## 2.2 Taxonomy of GNSS Threats: Interference vs Deception

An important feature of GNSS security research is the distinction between denial of service (jamming) and deception (spoofing and meaconing). While all may fall under the category of radio frequency interference, they have different characteristics and detection needs. Jamming attacks typically seek to degrade signal quality or loss of lock, and coincidentally show obvious power excursions, loss of lock or abrupt degradation of receiver features. Deception attacks seek to maintain plausible receiver behaviour while communicating false information such as timing and position, and are more difficult to detect using simple threshold techniques (Babić et al., 2025). Spoofing threats are further classified by complexity into three levels: GPS signal simulators, receiver-based spoofers that synchronize with authentic signals before re-broadcasting, and sophisticated multi-antenna systems capable of defeating spatial discrimination (Jafarnia Jahromi et al., 2012b). Modern detection frameworks must specifically address four distinct scenarios: high-power jamming, matched-power spoofing, overlapped signal interactions, and covered attacks where the spoofer masks all authentic reception (Broumandan et al., 2020).

### 2.2.1 Jamming and Coarse Spoofing: Detectability Under Strong Signatures

Simple spoofing attacks frequently employ simulator generated GNSS-like signals at higher power. Due to their poor timing alignment with the victim receiver's tracking state, they may lead to abrupt shifts in code phase, doppler frequency, correlator settings and power control. These attacks are often detected by AGC excursions, sudden changes of $C/N_o$, correlator asymmetry, implausible navigation solutions or loss of tracking stability (Shafique et al., 2021). Threshold and rule-based consistency checks are effective for this type of threat. A theory-driven review identifies that most of the modern research focuses on "Mimicking" attacks, where counterfeit signals copy the characteristics of authentic ones to avoid detection (Bertram et al., 2025). Multimodal detection can be utilized during the "searching process" of the acquisition module to detect intermediate spoofing almost in real-time across arbitrary signal intervals (J. Li et al., 2016).

### 2.2.2 Receiver-Informed Spoofing and Gradual Pull-Off

Spoofing attacks of a more sophisticated nature may try to match the code and Doppler estimates of the target receiver and may include smooth pull off approaches. These attacks minimise threshold-based violations and will often maintain tracking loop stability during the transition period. They are usually detected via multivariate feature analysis, sensor fusion or receiver self-consistency checks. The increasing availability of software defined radios and open GNSS software processing libraries makes it more accessible to perform receiver informed attacks (Islam, Bhuiyan, et al., 2024). This supports the need for detection approaches that can capture weak but correlated anomalies, rather than a single feature.

### 2.2.3 Meaconing: Replay-Based Deception with Authentic Content

Meaconing is a replay attack in which the attacker acquires authentic GNSS RF signals and retransmits them with a deliberate delay. Meaconing retains many of the authentic characteristics of the signal, including the navigation data bits, and possibly cryptographic authentication bits contained in the transmitted signal structure (Marnach et al., 2013; Seco-Granados et al., 2021). This produces a unique threat profile: content may be authentic while timing integrity is violated. Consequently, message authentication can not necessarily prevent replay, since replay is based on the difference between "authenticity of content" and "integrity of time of arrival". Meaconing is of special concern for aviation, as it affects position and quality information broadcasted through automatic dependent surveillance-broadcast (ADS-B). Research indicates that meaconing can cause dangerous position shifts in aircraft receivers, and so the definition of procedures is important for aviation safety (Steiner et al., 2024).

### 2.2.4 Why Meaconing is Especially Difficult for Traditional Detection

Meaconing is challenging to detect due to its plausibility at a normal $C/N_0$ ratio and lack of power anomaly indications (Semanjski et al., 2020). A carefully executed replay delay

applied commonly across satellites can appear largely as a common-mode clock bias in the receiver solution, preserving internal navigation consistency. This constrains the effectiveness of high-level solution tests unless more sophisticated checks or external references are used (Marnach et al., 2013). Therefore, the most actionable detection signal may occur in the receiver's internal dynamics during attack onset and takeover, when the composite of authentic and replayed signals perturbs tracking-loop behaviour.

### 2.2.5 Meaconing Detection as One Layer in a Defense Stack

The literature increasingly frames GNSS security as layered defense. Tracking-loop ML detection is not a replacement for spectrum monitoring, multi-antenna discrimination, inertial cross-checking, terrestrial timing references, or system-level mitigation logic. However, telemetry from the tracking loop provides receiver internal anomaly information that is functional even without additional sensors, thus making it applicable to embedded systems or mobile platforms that have limited hardware extension abilities (Siemuri et al., 2021). Such a multilayer view supports the thesis focus that is a telemetry-based detector that is computationally practical and insensitive to variation.

## 2.3 Receiver Tracking Loops and the Origin of Cross-Satellite Variance

One of the major issues with the use of ML-based GNSS interference detection is that the features do not evenly appear among the satellites in nominal conditions. Different PRNs are received under varying geometry, elevation, propagation conditions and varying dynamic conditions. This provides orderly baseline variability in internal telemetry including $C/N_o$ statistics, discriminator outputs, doppler estimates and correlator-shape metrics. These resulting distributions vary across PRNs and drift with time because of geometry changes and environment, that is why variability of features cannot be considered purely random noise (Chen et al., 2022; Li et al., 2025).

### 2.3.1 Geometry, Elevation, and Antenna Pattern Effects

Elevation angle may influence path loss, atmospheric attenuation, and antenna gain. High-elevation satellites typically present stronger and more stable received signals, whereas low-elevation satellites experience longer atmospheric paths and may fall in lower-gain regions of the antenna pattern. In many environments, masking and partial obstructions further degrade low-elevation channels. These factors produce PRN-dependent baseline differences in $C/N_o$ mean, $C/N_o$ variance, and tracking stability (Chen et al., 2022). Because elevation varies over time for each PRN, the baseline also drifts within the same satellite track.

### 2.3.2 Multipath Susceptibility and Correlator Distortion

Multipath is more severe for low-elevation satellites because reflections from ground and nearby structures become more likely. Multipath distorts correlator shapes, changes discriminator behaviour, and increases code tracking jitter. This produces PRN-specific correlator distortion baselines and variance patterns that can dominate the feature space. For replay detection, this is problematic because correlator distortion is also an informative meaconing indicator: the challenge is to distinguish attack-induced distortion from environment-induced distortion. A tri-layered neural network using average power and distortion correlation features can effectively classify received signals as interference-free, multipath, jamming, or spoofing with accuracy (Yakkati et al., 2022). Furthermore, combining multiple signal quality monitoring (SQM) metrics into a composite detector using probability of false alarm (Pfa) logic provides higher detection robustness against varying relative carrier phases than individual metrics (Sun et al., 2018).

### 2.3.3 Doppler and Doppler-Rate Differences Across Satellites

Relative motion between the receiver and each satellite yields different Doppler and Doppler-rate distributions. Carrier and frequency tracking loops operate in regimes that depend on these dynamics, affecting the baseline variance of doppler estimates and jitter metrics. Consequently, doppler-related features are naturally PRN-dependent, and

drift with satellite motion and receiver motion over time. This again creates a potential shortcut for ML models if evaluation splits do not enforce generalization.

### 2.3.4 Receiver Implementation Effects and Channel-Specific Behaviour

The distributions of tracking telemetry are influenced by receiver design decisions including loop bandwidth, discriminator design, filtering, quantization, estimation designs, and channel management. The same theoretical feature (e.g., code jitter) can vary in implementation. This is important in ML, as models can learn receiver specific artefacts, and features can become correlated with PRN identity due to channel dependent processing properties. Due to this reason, the literature stresses that, robust methods should consider the variability of satellites and receiver artefacts as first-class design constraints, as opposed to nuisance effects (J. Li et al., 2025).

## 2.4 Classical Detection Methods and Their Limitations Against Replay Attacks

Classical receiver-side GNSS spoofing and meaconing detection techniques are often threshold-oriented or detection of changes in the observable receiver characteristics such as a reduction of mean $C/N_0$, AGC spike, variance of the discriminator, asymmetry of the correlator, or Doppler-frequency anomaly (Akos, 2012; Jafarnia-Jahromi et al., 2012a; Psiaki & Humphreys, 2016; Semanjski et al., 2020). These techniques are still useful when attacks cause significant power variations or disrupt the tracking loops in an evident manner. However, there are two limitations that are especially relevant in replay-style interference. Global thresholds are difficult to tune because of the baseline variability across satellites. The threshold which could alert an anomaly on one PRN could be too tight on another or too loose on a third, creating false alarms or missed detections. Second, meaconing requires reception, delay and retransmission of actual GNSS signals, and data from the actual GNSS signal may still exist even if the navigation data integrity is compromised (Marnach et al., 2013; Semanjski et al., 2020). That means you are likely to see attack signatures that look like "relative excursions" from a channel's recent

baseline, not "absolute excursions" from one of the global boundaries. These restrictions lead to the approach taken in this thesis, namely comparing each satellite channel against its own set of baseline behaviour in a tracking-loop which includes PRN-relative tracking-loop features. As a result, the $C/N_o$ is considered as one of the receiver-side indicators, but is not used as an exclusive criterion for meaconing detection; the other included features may be DLL, code-phase, doppler jitter and correlator-based distortions (Semanjski et al., 2020; Zhu et al., 2022).

Mathematical descriptions have shown that meaconing provokes a quantifiable deformity in conventional estimators such as the moment method (MM) and the narrowband-wideband power ratio (NWPR). The difference between the estimated and actual $C/N_o$ is a sensitive signal-level signature that can be used to detect (Ghizzo et al., 2024). Additionally, strong variants of replay, known as distance decreasing (DD) attacks, can shorten pseudo-range measurements without altering the navigation message. These "authentication transparent" attacks require hypothesis testing, such as a goodness-of-fit (GoF) test on the prompt correlator output, to identify the subtle distribution changes they induce (K. Zhang et al., 2022). Absolute power monitoring is a more effective deterrent than simple $C/N_0$ monitoring because it forces the adversary to operate within an extremely narrow and predictable power window (Jafarnia Jahromi et al., 2012a). In the PRN code domain, subspace projection techniques allow single-antenna receivers to mitigate spoofing by projecting the received signal onto the orthogonal null space of the counterfeit signals (Han et al., 2017).

## 2.5 Machine Learning for GNSS Spoofing and Meaconing Detection

ML is attractive for GNSS deception detection because tracking-loop telemetry is multivariate, nonlinear, and context-dependent. Attacks may perturb multiple weak indicators simultaneously such as small increases in code-loop instability alongside subtle correlator distortions, where no single feature crosses a fixed threshold. ML can learn interactions among these features and build multivariate decision boundaries that are more sensitive than independent rules (Siemuri et al., 2021; Semanjski et al., 2020). A structured comparison of prior machine learning-based detection methods is provided in

Table 1, highlighting the datasets, extracted features, and validation protocols used in recent literature.

**Table 1** Comparison of Prior Machine Learning-Based GNSS Spoofing and Meaconing Detection Studies.

| Author & Year | Primary Data Used | Features Extracted | Validation Protocol | Cross-Satellite Generalization Tested? | Methodological Gap |
|---|---|---|---|---|---|
| **(Semanjski et al., 2020)** | Real-world Meaconing & Spoofing (Septentrio PolaRx5TR) | Multi-domain tracking-loop (Lock time, $C/N_0$, Clock drift) | Randomly Disruptive 70/30 split | No | Relies on random mixing of windows; susceptible to evaluation leakage from PRN-specific baselines. |
| **(Shafique et al., 2021)** | Simulated UAV sensor data | Signal characteristics (Jitter, Shimmer, Frequency modulation) | K-fold cross-validation & voting | No | Highly accurate for specific hardware but results are constrained by static PRN feature distributions. |
| **(Chen et al., 2022)** | TEXBAT & OAKBAT | multi-stage features (SQM-MV, $C/N_0$, PVT residuals) | Randomly Disruptive 70/30 split | No | Single-parameter methods are replaced by multi-parameter SVMs, but PRN-dependent variance is not addressed. |
| **(Aissou et al., 2022)** | SDR-collected GPS satellite signal features with simulated simplistic, intermediate, and sophisticated spoofing attacks | C/N0, pseudoranges, carrier phase, carrier Doppler, receiver time, early and late correlator outputs | Comparative supervised ML evaluation of five instance-based models, with Nu-SVM reported as the best-performing model | No | Uses receiver-level GNSS observables and correlator-related features, but does not explicitly test PRN-disjoint, LOPO, or satellite-independent generalization. |
| **(Yakkati et al., 2022)** | Synthetic multi-correlator GNSS signal data | Average power and correlation distortion features from | Classification performance evaluation using accuracy | No | Classifies multiple GNSS signal conditions, but does not explicitly evaluate PRN-disjoint or satellite- |

| | | | | | |
|---|---|---|---|---|---|
| | | multi-correlator GNSS receiver outputs | and confusion matrix | | independent generalization. |
| **Proposed Thesis (2026)** | FGI Spoofing Repository (GPS & Galileo) | 7 Statistical Features (Optimized Top 5 set) | Blocked Chronological, multiple PRN split validation, PRN-Disjoint | YES | Methodological Pivot: Introduces Local PRN-Relative Normalization to reduce the Generalization Gap. |

### 2.5.1 Supervised Learning: SVM and Random Forest

Supervised learning is widely used in spoofing and interference detection because controlled experiments can provide labelled clean and attacked intervals. SVM have been used to classify spoofing using engineered feature vectors and have demonstrated strong performance when class separation can be expressed through margin-based decision boundaries (Chen et al., 2022; Panice et al., 2017). Their strengths include stability in moderate dimensional feature spaces and relatively predictable behaviour. RF models are also widely used because they can model nonlinear interactions between features and give feature importance diagnostics, which aid interpretability in safety critical settings. RF is able to exploit the interaction between DLL instability, correlator distortion, and doppler instability in the transitional attack phases, frequently enhancing detection with more complex intervention patterns (Zarrinnegar et al., 2025). However, SVM and RF are generic supervised classifiers and do not intrinsically correct for cross-satellite variance. Without PRN-aware preprocessing and validation, they may take advantage of PRN-dependent receiver-observable baselines instead of attack-specific changes, which is a known risk in machine-learning studies where leakage or spurious correlations are present(Kapoor & Narayanan, 2023). This is particularly important when considering the data from GNSS, as receiver features, like $C/N_o$, Doppler behaviour, and tracking-loop statistics, can change from satellite to satellite and from condition to condition (Semanjski et al., 2020). SVM models trained on high-dimensional signal attributes including jitter, shimmer, and frequency modulation, have proven effective for protecting

UAVs from malicious signal injection (Shafique et al., 2021). Comprehensive multi-parameter detection involving nine distinct feature types (e.g., SQM moving variance, clock offset, and velocimetry residuals) significantly improves the F1-score of SVM classifiers on standardized datasets like TEXBAT and OAKBAT (Chen et al., 2022). Pairing c-support vector machines (C-SVM) with principal component analysis (PCA) enables the detection of even subtle programmed clock divergences by capturing the most statistically significant GNSS measurement variations.

### 2.5.2 Feature Engineering vs End-to-End Deep Learning in Embedded GNSS Security

Despite the fact that deep learning has been noted to be widely applicable in most areas of detection, in GNSS spoofing/meaconing detection studies, feature centric research tends to be pragmatic in nature. Advanced spoofing labelled datasets are limited; embedded platforms have computing and determinism constraints; interpretability is important because operators need to know why an alert was raised and what to do. Features like DLL instability, doppler jitter, correlator distortion map to physically meaningful receiver behaviours and hence enable engineering trust and debugging. (Zarrinnegar et al., 2025). Centralized systems such as the anti-jamming and meaconing system (AJAMS) suggest location stamps to create a huge global knowledge base of attack signatures, which is effectively using location authentication as an additional security consideration to GNSS terminals (Bull, 2010). Dual receiver correlation has been shown as one of the strongest defenses on the hardware side. Through cross correlation of unknown encrypted military signals at two dissimilar receivers, a system is capable of protecting publicly known civilian codes and identifying an assault within a crucial 1.2 seconds, which is an essential speed in modern day circumstances (Psiaki et al., 2013). Recent advancements in deep tabular learning, specifically the DeepGBM architecture, have achieved up to 98% accuracy in classifying malicious ADS-B messages and identifying specific attack types (Ahmed et al., 2025). To address the "zero-day" attack problem, representation learning models using a combination of variational autoencoders (VAE) and GANs can learn the distribution of genuine signals to identify novel, subtle spoofing vectors (Iqbal et al., 2024). Additionally, the least absolute shrinkage and selection operator

(LASSO) can de-compose correlation profiles into a dictionary of triangle-shaped replicas to identify spoofer peaks at sub-chip offsets as small as 0.2 chips. This thesis is aligned with deployment-oriented direction as it uses a small and intuitive feature set, developed from the outputs of the receiver tracking loops. All the selected features such as $C/N_0$, DLL discriminator behaviour, code-phase jitter, Doppler jitter and correlator distortion are already implemented or derivable in a software receiver for a GNSS. The proposed method does not rely on raw I/Q deep learning, external sensors, antenna arrays, nor heavy processing, thus is more practical for receiver-side implementations. Formulating spoofing detection as a multi-class reinforcement learning problem via deep q-networks (DQN) allows an agent to learn adaptive detection policies that are beyond static supervised models (Noman Chowdhury et al., 2026). The isolation forest algorithm provides a low-complexity, unsupervised method for detecting tampered observations in satellite files without requiring pre-labelled training data (Zuo et al., 2021).

## 2.6 The Generalization Gap: Cross-Satellite Leakage and Deployment Reality

The common limitation of ML-based GNSS detection is the discrepancy between reported benchmark accuracy and actual deployment performance. Random window-based train/test splits are used in many studies, which may unintentionally include samples of the same PRNs in both the training and test sets. In cases where cross-satellite baselines are highly varying, it is possible to use models to learn PRN-specific feature signatures as opposed to attack-induced deviations. This produces an evaluation leakage issue: the accuracy seems to be good when being tested but fails on the really unseen PRNs. In ML, the test distribution is not out-of-domain enough as compared to training, and thus the evaluation does not test the generalization required upon deployment (J. Li et al., 2025). PRN variation is a natural consequence of deployment conditions where each satellite is received with a different geometry, elevation, antenna-gain, propagation and multipath condition. Because of these factors, receiver observables like $C/N_0$, Doppler estimates, DLL discriminator behaviour and correlator metrics have different nominal baselines as explained in Section 2.3. The observable satellite set varies over time,

space, masking and dynamics. Detectors are thus required to generalize between satellites, geometry and environmental conditions. A model based on memorized PRN baselines is operationally unreliable despite a good result in a convenient benchmark split.

## 2.7 Feature Processing for Satellite-Independent Detection

For satellite independent GNSS meaconing detection, feature processing plays a crucial role due to the fact that not all of the GNSS receiver features are comparable for all satellites. Receiver-level features like $C/N_o$, code measurements, Doppler jitter, and tracking-related measurements have been employed in previous studies for detecting spoofing and meaconing (Psiaki & Humphreys, 2016; Semanjski et al., 2020). However, signal-quality indicators such as $C/N_o$ can vary depending upon the satellite elevation, receiver hardware, antenna characteristics, multipath, and environmental conditions (Paziewski et al., 2019; Y. Li et al., 2022). This thesis thus uses per-PRN baseline normalization prior to classification. All features are represented as deviations from a recent cleaned baseline, instead of absolute values, of the same PRN channel. This reduces the satellite-specific dependence of the input features and emphasizes changes due to attacks. Following the same processing step, this is also implemented in this thesis to mitigate the risk of learning PRN-dependent shortcuts instead of meaconing signatures, which is in line with broader machine-learning concerns regarding leakage and non-generalizable shortcuts (Kapoor & Narayanan, 2023).

## 2.8 Label Quality and Attack Onset Ambiguity in Experimental Data

An effective challenge to supervised detection is label alignment. The time of attack start recorded in experiments can be different from the time at which replay begins to affect the receiver tracking features. Also, there could be some variations in the response of the features and channels in terms of timing. This is similar to time-series anomaly detection tasks, wherein the abnormal behaviour can manifest itself in an interval instead of at a point (Tatbul et al., 2018). In GNSS meaconing experiments, attack beginning and ending are also detected through receiver behaviour over time, rather than only through

the experimental schedule (Marnach et al., 2013). If experiment scheduled timestamps are employed exclusively as a basis for labelling, then there is potential for the windows of transition to have incorrect labels, which interferes with model training and complicates evaluation of detection-delay based on labels.

An automated change detection process can be used to enhance the quality of labels by approximating onset based on feature dynamics. Instabilities of the DLL, doppler jitter, and correlator distortion are also features that tend to be sensitive to replay onset. Consensus-based change detection, using multiple features to reduce false triggers, provides a reproducible and objective onset estimate that supports supervised learning and operational evaluation of detection delay (Semanjski et al., 2020;Safi, 2025). This idea also strengthens methodological rigor because it reduces subjective manual labelling. This idea also strengthens methodological rigor because it reduces subjective manual labelling. The inclusion of a multipath estimating delay-lock loop (MEDLL) within a tightly coupled INS/GNSS integration facilitates the detection and suppression of spoofing components within 1 second of onset (Shang et al., 2022). A spoofing detection factor can also be derived by analysing changes in the covariance eigenvalues of a detection matrix formed by position vectors from multiple receivers (Jiang et al., 2018).

## 2.9 Literature Gap and Thesis Positioning

The literature establishes several key points. First, meaconing is a serious replay-based threat because it can preserve authentic signal content while corrupting timing integrity (Marnach et al., 2013; Seco-Granados et al., 2021). Second, tracking-loop telemetry provides informative signals for detecting deception, and ML can exploit multivariate feature interactions to detect subtle attacks (Semanjski et al., 2020; (Chen et al., 2022). Third, many ML studies report high classification accuracy but do not fully stress-test cross-satellite generalization due to evaluation protocols that leak PRN baseline information across training and test sets (J. Li et al., 2025). Sparse signal processing frameworks that employ novel linearization of the measurement can support detection and mitigation in stationary receivers against joint attacks targeting time, position, and velocity simultaneously (Lee et al., 2023).

The methodological gap is therefore not primarily the absence of classifiers, but the absence of leakage-conscious evaluation and satellite-independent feature representation. Operational GNSS receivers experience changing satellite sets and drifting baselines, so detectors must generalize across PRNs without relying on absolute baseline magnitudes. This thesis addresses that gap through an integrated design: (1) compact and physically interpretable tracking-loop feature engineering, where each selected feature has a direct receiver-level meaning, such as signal strength, DLL behaviour, code-phase stability, Doppler jitter, or correlator distortion, (2) data-driven attack onset labelling using consensus change detection, (3) local per-satellite standardization to reduce PRN baselines differences before classification, and (4) strict cross-satellite validation protocols that measure PRN-independent generalization. The contribution is not limited to improving classification accuracy, but it also includes a deployment aligned evaluation of meaconing detection performance.

# 3 Research Method

## 3.1 Methodological Objectives and Design Logic

This thesis proposes a machine-learning system to detect GNSS meaconing attacks using receiver tracking-loop telemetry. A key constraint on the methodology is that the detector must be satellite-independent: it must perform well on PRNs not encountered in the training data. This is necessary because PRNs are constantly changing in practice. As a result, the methodology is designed to (i) avoid evaluation leakage (where models learn to detect PRNs behaviour rather than attacks), and (ii) create a satellite-independent feature representation that is implementable in real time. The work follows a staged experimental design:

1. Convert high-rate tracking telemetry into compact, interpretable statistical features over fixed windows.
2. Estimate attack onset via change detection to reduce labelling ambiguity and to support realistic detection delay measurement.
3. Establish baseline supervised models (SVM and RF) and interpret feature contributions.
4. Stress-test true generalization using strict PRN-holdout and leave-one-PRN-out (LOPO) validation, then address failure using satellite-relative normalization.
5. Quantify operational behaviour (time-to-detection (TTD), assess robustness to baseline window choice, and discuss calibration and deployment feasibility.

All implementation was performed in `MATLAB R2024b` using standard `Statistics and Machine Learning Toolbox version 24.2` functions for partitioning, training, and evaluation.

## 3.2 Data Source and Telemetry Structure

### 3.2.1 Recording characteristics

The data analysed in this study originate from FGI's GNSS Spoofing Dataset Repository (Islam et al., 2024b), which provides raw I/Q recordings of live GNSS signals under several spoofing scenarios. In this thesis, the Meaconing DFMC scenario has been used, specifically L1/E1 recording, which has GPS L1 and Galileo E1 signals. The dataset paper reports the data duration of around 478 s in `MCD_L1_E1.dat` file and the duration of the telemetry at 1000 Hz in the processed tracking output used in this thesis is around 480 s. The scenario of the LabSat 3 Wideband record-and-replay device was utilized in the Meaconing DFMC scenario. The meaconing signal was captured at a distance of about 60 m around the rooftop antenna and then replayed along with the live signal with a delay of about 15 minutes. The receiver remained at rest during the recording, and the signal was replayed after 155 s of clean recording. The initial 130 s of the dataset is reported to be free of intentional interference, which is why an early pre-attack baseline is used to normalize the data. The raw I/Q recordings were obtained with NSL stereo dual-band GNSS front-end. The recording setup of the L1/E1 band used the following parameters: center frequency of 1569.03 MHz, sampling rate of 26 MHz, 8-bit real samples and bandwidth of 4.2 MHz. The documentation of the dataset does not present a directly usable numerical replay power ratio of the meaconing signal in the analysis that has been done herein. Thus, the receiver tracking-loop telemetry is viewed as the primary source of evidence and the assessments of the receiver-observed effects of the replay signal as opposed to the modelling of attacker-side power parameters.

The raw I/Q recorded data were processed with the `FGI-GSRx` software-defined receiver (Islam et al., 2024a), which generated the `trackData` files containing the tracking-loop telemetry that is being analysed in this thesis. The identical feature extraction, windowing and evaluation pipeline were individually applied to the GPS L1 and Galileo E1 `trackData` files. The GPS analysis with 14 PRN channels and the Galileo analysis with 8 PRN channels were used in the final pipeline. PRN identifiers are thus inter-read in their

respective constellation. The available PRNs across the processed GPS and Galileo files were: 1, 2, 3, 7, 8, 10, 12, 13, 14, 15, 17, 19, 21, 22, 23, 24, 25, 32, and 33.

### 3.2.2 Per-sample features and special fields

The per-sample fields used for windowed feature extraction are:
CN0fromSNR, meanCN0fromSNR, noiseCNOfromSNR, codePhase, dllDiscr, doppler frequency.

Two additional structures are present:

1. `trackState` has length 19 and is not aligned per sample. It is not used in the main per-window feature table because the pipeline is based on per-sample-to-window aggregation.
2. `mulCorrFingersOut` is a multi-correlator output matrix of size 480000 × 17 for each channel. The last feature of distortion of the correlator does not apply solely to four fixed taps. Instead, it uses the prompt finger index stored in the receiver output and compares the prompt magnitude against the total non-prompt side energy across the available correlator fingers. This eliminates the need to have a manually assumed prompt location and corresponds to the implemented feature extraction code.

### 3.2.3 Why tracking-loop telemetry is a valid basis for meaconing detection

Meaconing preserves authentic navigation data content and can maintain plausible signal appearance. In such conditions, detection signals are often more apparent in receiver internal dynamics (code and carrier tracking stability, discriminator behaviour, correlator structure) than in message level validity. The methodological implication is that telemetry-based detectors can remain viable even in constrained receivers where external sensors or multi-antenna techniques are not available. This thesis therefore treats tracking-loop telemetry as the primary evidence source and explicitly addresses the cross-satellite variance that naturally arises in those features. The entire data processing pipeline is illustrated in Figure 2.

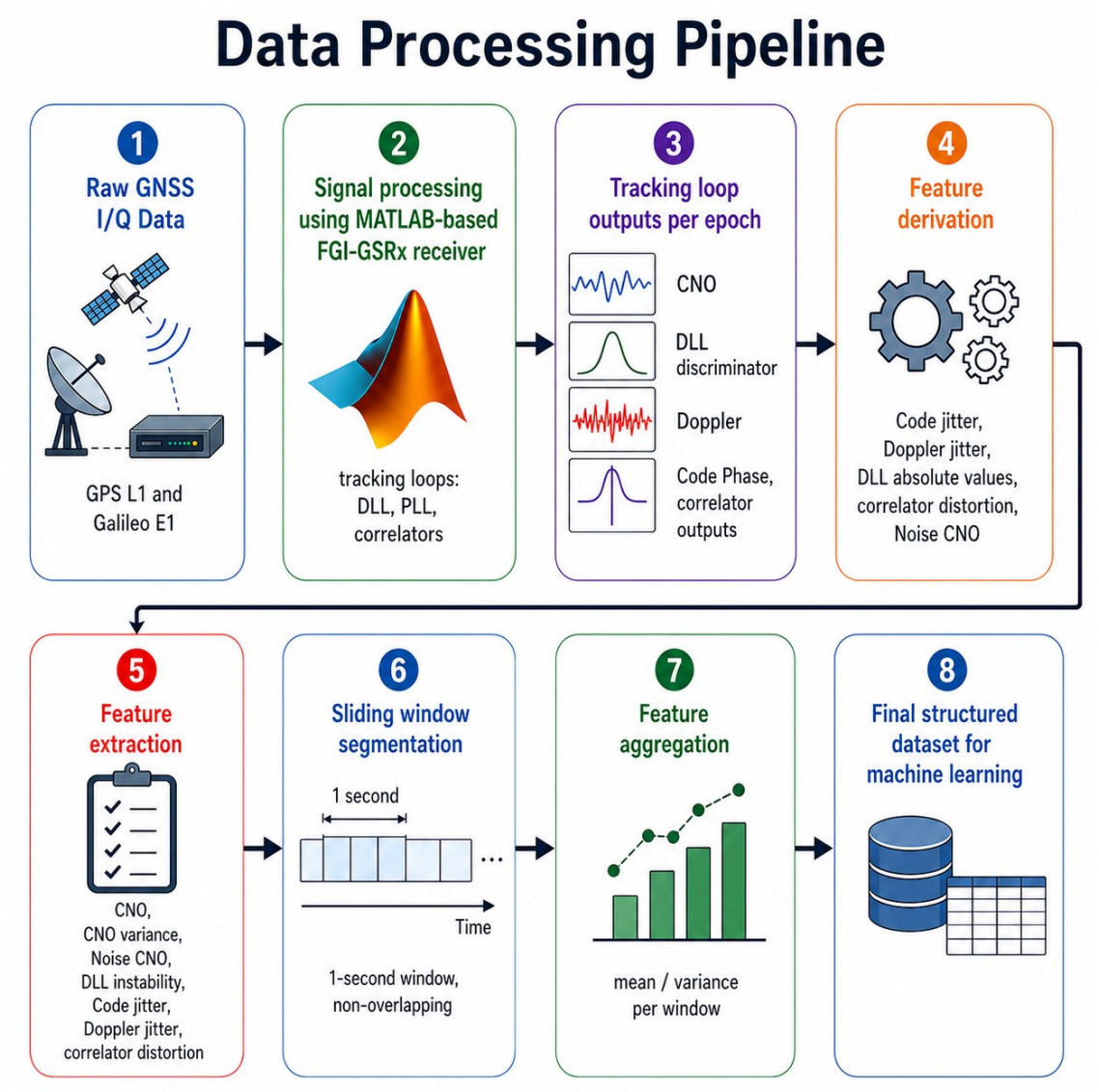


**Figure 2** Data Processing Pipeline Overview. NB: This figure was created using AI. The full prompt used for generation is provided in Appendix 1 (OpenAI, 2026).

## 3.3 Windowing Strategy and Dataset Representation

### 3.3.1 Rationale for windowing

The raw sampling rate (1000 Hz) produces high-dimensional time series that are not suitable for straightforward supervised learning without substantial sequence modelling overhead. Windowing compresses raw telemetry into compact statistics that preserve stability and jitter behaviour while enabling interpretable learning and controlled

evaluation. Windowing also permits deterministic alignment between time, labels, and evaluation metrics such as detection delay.

### 3.3.2 Window specification and feature table layout

In both constellations, the telemetry is divided into non-overlapping (1 second) windows. There are 1000 samples in each window. This results in about 480 windows per PRN over the processed recording. In the final pipeline, there are 6720 windows in the GPS dataset among 14 PRNs, and 3840 windows in Galileo dataset among 8 PRNs. The rows are associated with a single PRN-specific time window and have metadata fields alongside the seven extracted features and the class label.

## 3.4 Feature Engineering and Physical Rationale

Feature design follows two constraints simultaneously:
First, physical interpretability: each feature should map to a tracking-loop behaviour that can plausibly be perturbed by replay overlap or takeover. Second, satellite-independence: the representation must be transformable into deviations from a local PRN baseline, because absolute feature magnitudes differ naturally across satellites due to geometry, elevation, antenna gain patterns, and channel operating conditions. This section defines the full feature set and then explicitly justifies why each feature can still be useful under PRN variability when paired with local normalization (Section 3.9).

### 3.4.1 Feature definitions (f1–f7)

Seven window-level features are computed for each PRN. Let w denote a 1-second window containing N=1000 samples. Let n be the sample index inside the window.
**f1: Mean C/$N_0$.** This feature is computed as the mean of CN0fromSNR samples within the window:

$$f_1(w) = \frac{1}{N} \sum_{n \in w} CN0[n] \tag{1}$$

**f2: C/$N_o$ variance.** This feature captures short-term signal-power variability within the window:

$$f_2(w) = Var(CNO[n]), \qquad n \in w \tag{2}$$

**f3: Noise C/$N_o$ statistic.** This feature is computed as the mean of the receiver noise-related C/$N_o$ estimate:

$$f_3(w) = \frac{1}{N} \sum_{n \in w} NoiseCN0[n] \tag{3}$$

**f4: DLL instability**. The per-sample DLL magnitude is first computed as:

$$DLL_{abs}[n] = |DLL[n]| \tag{4}$$

The window-level DLL instability feature is then computed as:

$$f_4(w) = Var(DLL_{abs}[n]), \qquad n \in w \tag{5}$$

**f5: Code phase jitter.** Code jitter is computed from consecutive code phase estimates:

$$CodeJitter[n] = |CodePhase[n] - CodePhase[n-1]| \tag{6}$$

The window-level feature is:

$$f_5(w) = \frac{1}{N} \sum_{n \in w} CodeJitter[n] \tag{7}$$

**f6: Doppler jitter.** Doppler jitter is computed from consecutive Doppler estimates:

$$DopplerJitter[n] = |Doppler[n] - Doppler[n-1]| \tag{8}$$

The window-level feature is:

$$f_6(w) = \frac{1}{N} \sum_{n \in w} DopplerJitter[n] \tag{9}$$

**f7: Correlator distortion metric.** This feature is derived from the multi-correlator output matrix mulCorrFingersOut. Let $C_i\,[n]$ denote the magnitude of the $i$ -th correlator finger at sample $n$, and let $C_p\,[n]$ be the prompt correlator magnitude, where $p$ is prompt finger index and $M$ is the number of multi-correlator fingers. The side energy is computed as:

$$SideEnergy[n] = \sum_{i=1}^{M} C_i\,[n] - C_p[n] \tag{10}$$

The correlator distortion metric is then:

$$CorrDistortion[n] = \frac{SideEnergy[n]}{C_p[n] + \epsilon} \tag{11}$$

where ϵ is a small constant used to avoid division by zero. The final window-level feature is:

$$f_7(w) = \frac{1}{N} \sum_{n \in w} CorrDistortion[n] \tag{12}$$

This definition measures the amount of correlator distortion by dividing non-prompt side energy by prompt correlator energy. It is thus more accurate than considering correlator distortion as a scalar summary of correlator shape which is not defined.

The seven features above were chosen since meaconing injects an additional signal component, which is a replay, into the internal tracking state of the receiver. The chosen features describe the complementary effects such as signal power variations, short-term C/N0 variations, noise behaviour, DLL instability, signal code and Doppler jitter, and correlator energy distortion. Together, they provide a compact representation of signal-quality changes and tracking-loop disturbances during meaconing.

### 3.4.2 Why these features are robust to PRN variability

Robustness here does not mean the raw feature distributions are identical across PRNs, they are not. The robustness comes from using features that remain meaningful after converting them from absolute magnitudes into PRN-relative deviations.

$C/N_o$ mean (f1) is strongly satellite dependent in absolute terms, but deviations from a PRN's own baseline can reveal abrupt or unusual changes in received signal composition. This is especially relevant for replay overlap, where a second signal component may subtly shift effective power or tracking behaviour.

$C/N_o$ variance (f2) is less dependent on absolute link budget and more related to short-term stability. Even when a PRN is naturally weaker, its variance profile in stable conditions tends to remain within a consistent range. Replay interference can increase micro-instabilities, making variance a useful deviation-based indicator.

Noise $C/N_o$ (f3) is typically expected to respond under interference like conditions. Although its usefulness may depend on the specific recording conditions and receiver behaviour.

DLL instability (f4) captures code tracking stress. While code tracking performance depends on geometry and multipath, the key is that a PRN's DLL variance baseline is often locally stable over short intervals. Replay onset can perturb discriminators due to composite correlation peaks or imperfect alignment between authentic and replayed components.

Code phase jitter (f5) is conceptually similar to DLL instability but expressed via dispersion in code tracking output rather than discriminator behaviour. These two features together capture both "error signal volatility," meaning rapid or irregular fluctuations in the DLL discriminator output, and instability in the resulting code tracking estimates.

Doppler jitter (f6) is naturally PRN dependent due to relative motion and receiver dynamics, but short-term deviations from a PRN's baseline can still capture tracking stress during transition periods.

Correlator distortion (f7) is theoretically attractive for spoofing and meaconing because replayed signals can alter the correlation structure. In practice, the advantage of this

feature depends on how strongly such distortion is expressed in the observed receiver telemetry.

## 3.5 Attack Onset Estimation via Change Detection

### 3.5.1 Motivation: label alignment and “attack schedule vs receiver response”

Supervised training requires labels, but attack onset in telemetry may not align precisely with external scenario timing. Mislabelling transitional windows can degrade training and distort TTD analysis. Therefore, attack onset is estimated from the telemetry using change-point detection, improving label precisely and repeatability.

### 3.5.2 Change-detection procedure and feature selection

Change detection was mostly achieved with $C/N_o$-related behaviour and DLL instability, since they displayed interpretable transitions around the onset of the replay. The other features, including Noise $C/N_o$, code jitter, Doppler jitter and correlator distortion, were also checked to ensure whether the attack induced consistent changes across multiple tracking-loop domains. Some of these $C/N_o$ and DLL-related change points were clustering around 155 to 164 s across satellites, whereas some early Doppler changes near the start of the recording were said to be due to receiver initialisation rather than the onset of an attack. On the basis of the clustering of the transitions observed by the receiver, and the reported scenario-timing, the consensus onset time to be used throughout the experiments is:

```
attackStartSec = 155 seconds.
```

This value is in agreement with the description of the dataset, where the retransmitted signal of meaconing is added after about 155 s of clean recording.

## 3.6 Labelling Strategy and Class Structure

Labels are assigned at the window level:

```
If tStartSec < 155, y = 0 (Normal)
```

```
If tStartSec ⩾ 155, y = 1 (Attacked)
```

For GPS:

The dataset contains 6720 windows total including 2170 normal and 4550 attacked windows.

For Galileo:

The dataset contains 3840 windows total including 1240 normal and 2600 attacked windows. The imbalance is preserved because it reflects the operational structure of an event that begins and persists, and it allows meaningful evaluation of false alarms and missed detections under realistic class proportions.

## 3.7 Pipeline Description in Pseudo-Code Form (Prose, Not Code-Heavy)

The pipeline was implemented in a step-by-step manner so that each stage could be checked and repeated. To start with, the GPS L1 and Galileo E1 trackdata files were loaded, and the necessary tracking fields were checked. The outputs of the receiver were then transformed into seven physically interpretable features. Then the data of each PRN was split into non-overlapping 1-second windows, with one feature vector constructed each PRN-window combination. It was determined that the attack onset would be 155 s, based on the change points observed by the receiver as well as the description of the dataset. This onset time was then used as window labels. To prevent any leakage, feature scaling and PRN-relative baseline normalization were only applied with training data or clean pre-attack data, depending on the validation setup. The SVM and RF models were trained following the appropriate train, validation and test splits. The last assessment consisted of blocked chronological validation, PRN-disjoint testing, multiple PRN split validation, LOPO validation, time-split validation, robustness analysis, and TTD analysis. This structure helps to keep the dataset preparation, labelling, training and evaluation phases well-organized and repeatable. The overall machine-learning workflow, from labelled feature dataset construction to validation, testing, and additional analyses, is summarized in Figure 3.

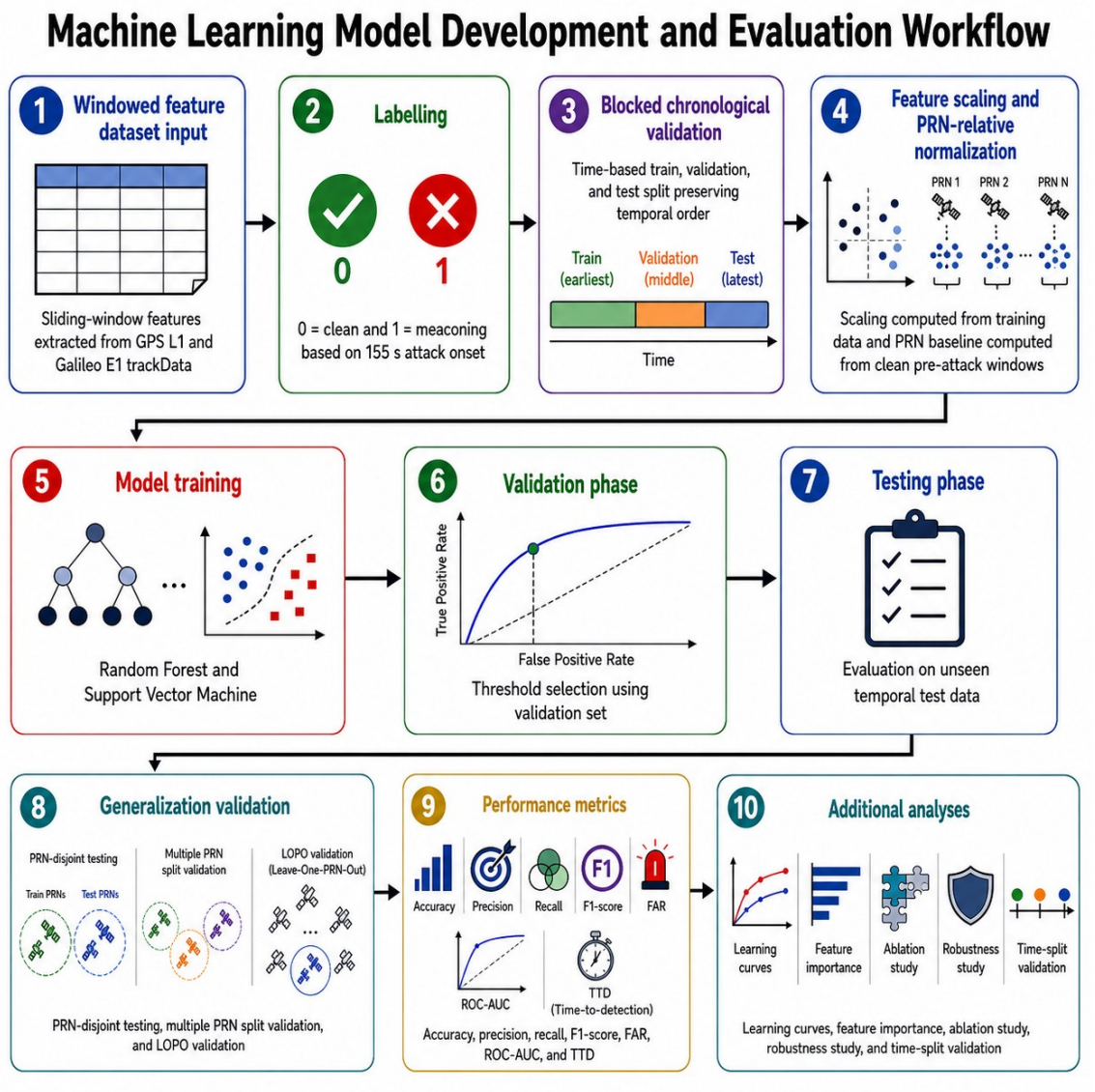


**Figure 3** Machine Learning Model Development and Evaluation Workflow. NB: This figure was created using AI. The full prompt used for image generation is provided in Appendix 2 (OpenAI, 2026).

## 3.8 Partitioning Strategy and Integrity Checks (No Leakage)

### 3.8.1 Why “random window split” is only a baseline

Random stratified split trains and tests windows independent of PRN identity. This may overestimate results because the model encounters the same PRNs in both sets, and PRN-specific baselines can serve as a crutch. For the purposes of this thesis, random

splits are used as an initial test to ensure that the features contain discriminative information.

### 3.8.2 Blocked Chronological Validation

A blocked chronological validation was carried out to determine if the classifier was able to generalize across blocks of time rather than benefiting from a random mixing of adjacent windows. Here, the sequence of windowed features was split into contiguous temporal blocks and training and testing were conducted on different blocks while maintaining temporal continuity. This approach eliminates temporal leakage between the windows and allows a more rigorous test of whether the decision boundary learned is valid across the temporal development of the recording, especially around the time when the signal is transitions from clean to attacked conditions.

### 3.8.3 PRN-disjoint splitting (cross-PRN evaluation)

The PRN-disjoint split assesses generalization between satellites by making sure that the test set has PRNs which are not contained in the training set. This protocol is important in the sense that when it is deployed it is observed that the visible set of satellites varies with time. When a model only works well when training and testing are done with the same PRNs, then the result might be due to satellite-specific baseline learning but not attack detection.

### 3.8.4 Multiple PRN Split Generalization Validation

In order to have a statistically significant estimate of satellite-independent performance, multiple PRN split validation was also carried out. In this validation strategy, the available PRNs from both GPS and Galileo were repeatedly divided into training and testing groups. The classifier was trained on a subset of PRNs and tested on entirely novel PRNs in the rest of the subset. This is unlike a single PRN-disjoint split since it is not based on a single train-test satellite split. Rather, the process is repeated under the multiple random PRN

combination which will allow the evaluation to measure the average performance as well as the variability of the performance across the subsets of the satellite.

This thesis evaluates 20 independent PRN-based in which training was done on 70% of the combined GPS and Galileo PRNs, and the remaining 30% was held out to be tested. To prevent the confusion of GPS and Galileo satellites having the same number PRN identifier, Galileo PRNs were internally offset during implementation. Accuracy, precision, recall, F1-score, false alarm rate, and AUC were calculated using each split. The 20 splits of the final results are reported in Table 6 as mean standard deviation across the 20 splits. This validation is a more powerful measure of generalization than a unitary holdout split because it provides information about whether or not the detector is stable across varying combinations of satellites. It also shows whether certain PRN subsets are more challenging than others which can be important to GNSS because a satellite signal is naturally affected by geometry, elevation, multipath, and receiver tracking behaviour.

### 3.8.5 Leave-one-PRN-out validation

LOPO validation is being carried out as a supporting validation technique. The training of the model in LOPO is on all PRNs except one and tested on the omitted PRN. This gives per satellite diagnostic data and assists in identifying satellite unique cases like threshold miscalibration. Nevertheless, since LOPO tests one satellite at a time, the multiple PRN split validation is the more generalized test as it tests multiple, unseen satellites simultaneously across repeated random splits.

### 3.8.6 Leakage-prevention integrity checks

To ensure the PRN-disjoint protocol is valid, the methodology includes several integrity checks:

**Check 1:** PRN set disjointness

Verify intersection `(PRN_train, PRN_test)` $= \emptyset$.

**Check 2:** Baseline normalization uses only “past” data

When computing PRN baselines for normalization, use only pre-attack windows (e.g., `tStartSec < 120 s`) so that post-attack information does not leak into feature scaling.

**Check 3:** Avoid window overlap between train and test

Non-overlapping windows remove ambiguity about shared samples contributing to multiple partitions (important for time-series leakage control).

These checks ensure the results correspond to cross-satellite generalization.

## 3.9 PRN Baseline Normalization (Satellite-Relative Deviation Modelling)

Since PRN baselines differ naturally, the detector should not use absolute feature magnitudes. Instead, each PRN's feature values are converted to deviations from that PRN's nominal baseline. This is the methodological pivot of the thesis: it enforces satellite independence by removing PRN identity encoded in absolute operating points.

### 3.9.1 Baseline computation and causal implementability

Baseline mean for each PRN is computed using windows in a pre-attack interval ending at baselineEndSec = 120 seconds (chosen to be strictly before the estimated attackStartSec = 155 seconds). Let $\mathbf{x}_{p,w}$ be the feature vector for PRN $p$ at window $w$, and let $\mathcal{B}_p$ be the set of baseline windows for PRN $p$, where $tStartSec < 120$s. The baseline mean is:

$$\mu_{p,base} = \frac{1}{|\mathcal{B}_p|} \sum_{w \in \mathcal{B}_p} \mathrm{x}_{p,w} \tag{13}$$

The PRN-relative feature vector is then computed as:

$$\tilde{\mathrm{x}}_{p,w} = \mathrm{x}_{p,w} - \mu_{p,base} \tag{14}$$

This approach is computationally inexpensive and causally implementable: in a live receiver, the baseline can be formed from past windows only.

### 3.9.2 Cross-PRN performance restoration

After baseline normalization, cross-PRN evaluation was repeated to assess whether satellite-independent generalization improved. This tests whether the attack signature is expressed more consistently as a deviation from a local PRN baseline than as an absolute feature magnitude.

## 3.10 Model Selection, Training, and Feature Reduction

### 3.10.1 SVM baseline

The SVM baseline uses an RBF kernel trained on standardized features with a stratified 70/30 holdout split. The baseline performance provides an initial indication that the feature space is separable, but it is not treated as evidence of satellite-independent generalization.

### 3.10.2 RF model and interpretability

The RF model is evaluated under the random split to observe improvements in false positive and false negative reductions relative to the SVM. Beyond accuracy, RF provides feature importance estimates, allowing the thesis to interpret which tracking behaviours are most informative.

### 3.10.3 Ablation study and “Top 5” feature set

Ablation analysis was used to evaluate reduced feature, and an optimized top 5 feature set [1, 2, 4, 6, 7] was selected. This streamlined set focuses on the most reactive indicators of meaconing, namely signal power trends and tracking loop volatility, while also reducing computational overhead.

### 3.10.4 Overfitting and Model-Complexity

To detect overfitting, we performed two types of analyses: a learning-curve analysis and a sensitivity-analysis on model complexity. The learning curves were used to assess the convergence of training and validation performance with increasing training dataset size. Meanwhile, the model-complexity analysis examined the generalisation gap as a function of increasingly flexible tree configurations, namely variation in minimum leaf size. A model was deemed overfit in the case of continued improvement in training performance without improvement in validation performance, or in the case of a systematic increase in the train-validation gap with increasing complexity. As a result, the final set of hyperparameters corresponded to the configuration that maintained high validation AUC, while minimising the generalisation gap, rather than the configuration that achieved the highest fit to the training data.

## 3.11 Thresholding, Calibration, and Why AUC Matters

In satellite-independent evaluation, threshold dependent metrics (accuracy, recall, specificity) can vary due to shifts in predicted probability distributions across PRNs. AUC is threshold-independent and measures ranking quality: how well the model separates attacked vs normal windows regardless of the cutoff. Therefore, AUC is emphasized when comparing across PRNs and across baseline normalization settings.

## 3.12 Time-to-Detection Analysis

TTD measures how quickly the detector identifies the attack after the estimated onset. This matters because replay-induced timing bias can propagate into dependent systems. The methodology computes detection delay at 1 second resolution, consistent with the windowing scheme. To reduce sensitivity to single-window noise, detection is declared only after K consecutive positive windows. TTD is evaluated using PRN baseline normalization ( $baselineEndSec = 120$ ), the Top 5 feature set [1, 2, 4, 6, 7], an RF classifier with 200 trees, and a stability rule of $K = 3$ consecutive positive windows.

## 3.13 Robustness and Sensitivity to Baseline Window Length

Baseline normalization depends on the length of clean pre-attack data used to estimate the PRN baseline. A method that only works for one carefully chosen baseline interval is fragile. Therefore, baselineEndSec is varied; in methodology baselineEndSec $\in$ {80, 100, 120} seconds.

## 3.14 Computational Complexity and Deployment Feasibility

### 3.14.1 Feature extraction cost

Feature extraction operates per PRN once per second. For each PRN window, computations are straightforward statistics (mean, variance, dispersion) over $N = 1000$ samples and a small set of correlator taps. This has complexity $O(N)$ per feature per window with small constants. In real time, the receiver already processes tracking loops at high rate; the additional cost is aggregating statistics once per second. PRN baseline normalization requires maintaining baseline means per PRN per feature and subtracting them from the current window vector. This has a complexity of $O(d)$ per window where $d$ is the number of features (5 or 7).

### 3.14.2 Model inference cost

RF inference is performed at a 1 Hz window rate. Using a configuration of 200 trees and a minimum leaf size of 10 (selected to prevent overfitting while maintaining model depth), the inference cost remains low. This suggests that multi-channel real-time monitoring may be practical, subject to implementation-specific benchmarking. SVM inference with an RBF kernel can be more expensive if the number of support vectors grows large. Both models are acceptable for offline evaluation; RF is preferred here due to strong performance and interpretability.

### 3.14.3 Memory and state requirements

A real-time deployment would store:
Per PRN: rolling window statistics accumulators (or last 1 second buffer), baseline mean vector, and recent K decision history for stability rule.
For the evaluated constellations and a 5-feature model, state is minimal (on the order of hundreds of floats). This supports the feasibility claim: satellite-relative normalization and 1 Hz inference are lightweight.

### 3.14.4 Determinism and interpretability

RF supports feature-importance analysis, aiding interpretation of which features contribute most strongly to detection. Additionally, the pipeline avoids deep sequence models, which often complicate real-time determinism and explanation. This supports the practical feasibility of the proposed detector.

# 4 Results and Empirical Evaluation

## 4.1 Data Characterization and Pre-processing Audit

Prior to testing the supervised models, the telemetry was inspected to confirm the presence of a stable pre-attack baseline for baseline estimation and justify the chosen labelling approach. Here, a stable baseline means that the tracking-loop features remained within their normal pre-attack range without sustained drift, abrupt structural change, or attack-like disturbance. This inspection was an essential part of the methodology because both PRN-relative baseline normalization and attack-onset labelling depend on the reliability of the pre-attack reference period. The proposed detector is intended to recognize abnormal departures from normal tracking behaviour. Consequently, if the nominal state is unstable, poorly defined, or inconsistent across channels, then any later classification result would be difficult to interpret. The dataset for GPS consisted of 14 tracked PRNs observed over a 480-second recording sampled at 1000 Hz, giving 480 windows per PRN, and 6720 windows in total. This yielded 2170 normal and 4550 attacked windows. The Galileo dataset consisted of 8 tracked PRNs observed over a 480-second recording sampled at 1000 Hz, giving 480 windows per PRN, and 3840 windows in total, yielding 1240 normal and 2600 attacked windows.

The first empirical question was whether the early part of the recording provided a sufficiently clean segment for baseline estimation. Examination of the noise-related and tracking-loop features suggested that the pre-attack interval was comparatively stable, particularly prior to the 155 s transition that was later confirmed through change detection. This stability justifies the use of a causal pre-attack baseline window for PRN-relative normalization. The selected value baselineEndSec = 120 s was therefore chosen with a deliberate safety margin. It remained well before the telemetry-supported attack onset and reduced the risk that transitional contamination would enter the baseline estimate. This supports the use of a causal pre-attack baseline for PRN-relative normalization in the present dataset.

The second part of the audit was to check whether the satellites tracked were at the same nominal operating point, they were not. The feature distributions in the pre-attack phase showed significant PRN-to-PRN differences, which translates to a distinct baseline behaviour of each satellite channel before the meaconing event. This was an affirmation that there would be no single global reference level that would suit all PRNs and that PRN-relative baseline normalization before model training and evaluation was needed. Absolute $C/N_0$, DLL variance, code phase jitter and doppler jitter were different for different PRNs, as is expected since satellite elevation, geometry, antenna gain, multipath and relative motion all contribute to channel specific effects. If the same PRNs appear in both the training and test partitions, the model may learn PRN-specific information encoded in the absolute feature values rather than attack specific variations. Therefore, the audit demonstrates that PRN-dependent baseline variation is not just a theoretical concern, but an observed property of the dataset. This is particularly evident in the DLL related observables. For instance, GPS PRN 32 exhibited nominal discriminator noisier than stronger channels in clean tracking conditions. This explains why using one global threshold across all satellite channel is challenging, since each PRN may have a different nominal feature's operating range before the attack. This suggests baseline should be viewed locally as stable for each PRN, rather than globally consistent across all PRNs. This point is examined further in the PRN disjoint validation analysis, where the use of absolute features is shown to reduce cross-satellite generalization.

The final issue related to pre-processing was label alignment. The scheduled attack start time does not always coincide exactly with the time at which receiver tracking features begin to respond. Thus, onset determination was made via telemetry-based change detection, rather than by accepting the external timestamp. DLL instability, doppler jitter, and correlator distortion were investigated for the different satellites, and a consistent onset of 155 s was adopted for the experiments. The onset of early change points in doppler at 0.5 s was dismissed as a receiver startup effect. This approach improves the validity of the labels that are used for supervised learning, and it also makes the TTD

analysis later more meaningful, because the model is evaluated relative to the onset time that is readily visible in the telemetry.

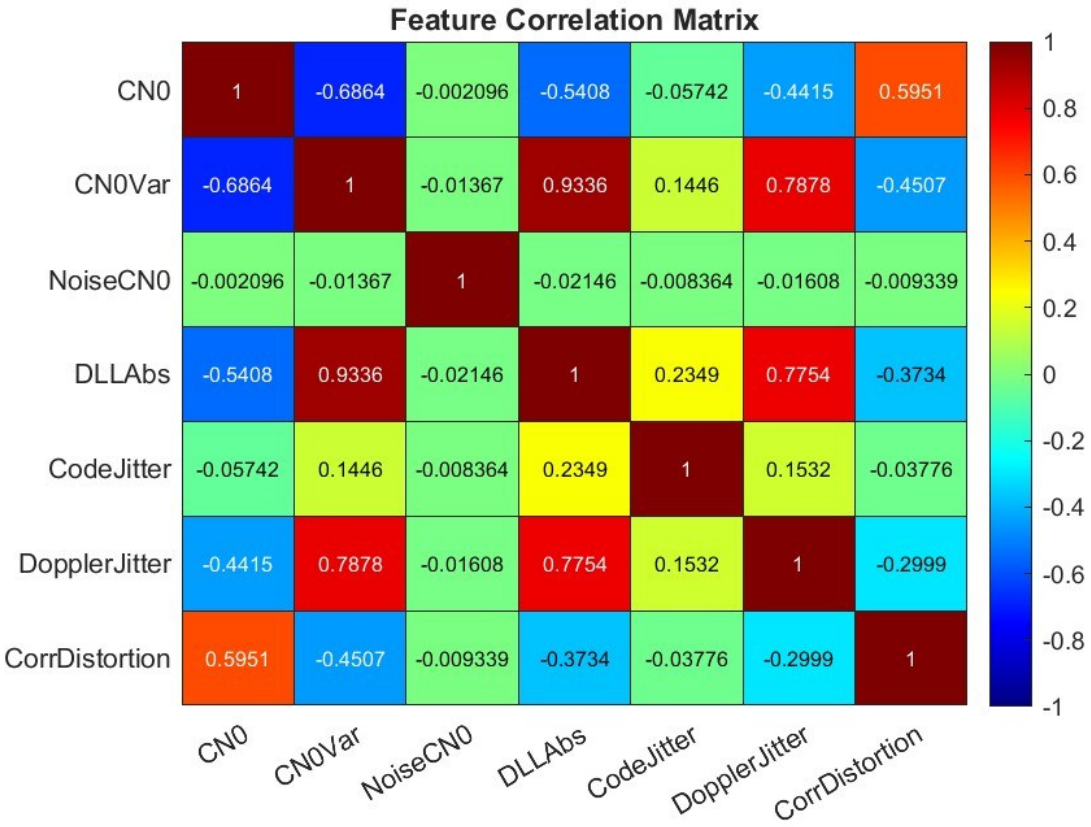


**Figure 4** Correlation heatmap of extracted GNSS tracking-loop features.

The feature correlation matrix in Figure 4 offers a descriptive look at the seven features extracted. The heatmap shows that features related to signal power are correlated, but some of the features related to tracking loop volatility are less correlated with signal

power. This supports the use of a multi-domain feature set in which complementary information is drawn from both signal power and tracking stability.

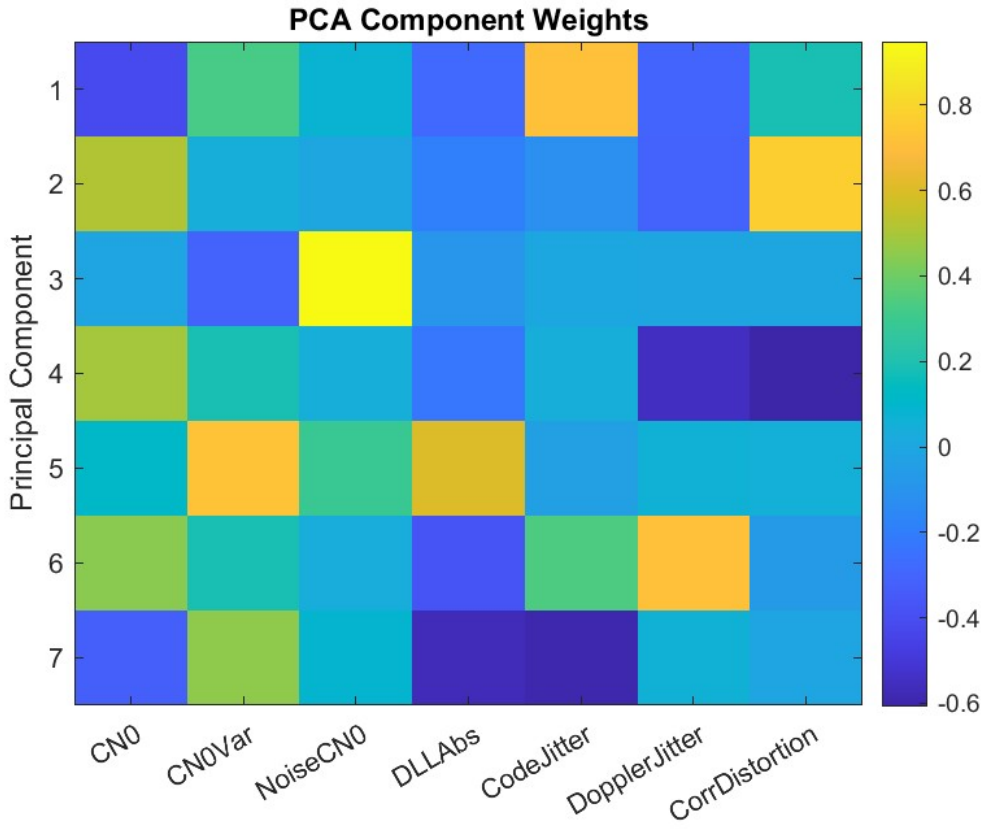


**Figure 5** Principal Component Analysis (PCA) Component Weights.

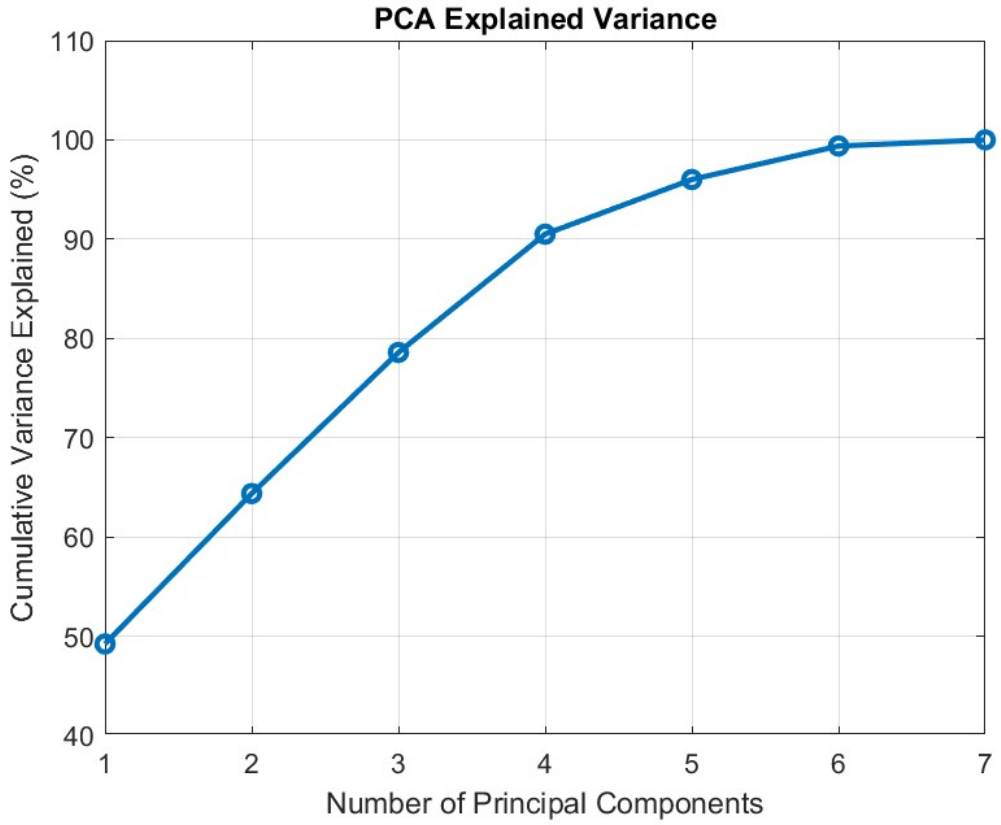


**Figure 6** Principal component explained variance analysis for evaluating dimensional structure in the GNSS tracking-loop feature space.

To understand the intrinsic dimensionality of the attack signature, a PCA was conducted. The cumulative explained-variance curve in Figure 6 shows that the first three principal components account for nearly 80% of the total variance, while the first four account for more than 90%, indicating that the data contains a structured but still multi-dimensional pattern. The component-weight heatmap in Figure 5 further illustrates the contribution of each extracted feature to the principal component. The result shows that the observed variance is distributed across multiple features rather than being dominated by a single feature, with contributions appearing across both energy-related and phase/frequency-related terms. Together, Figures 5 and 6 indicate that the attack signature is multi-dimensional rather than governed by a single dominant feature, which is consistent with the use of a multivariate classifier.

## 4.2 Benchmark Classification Results

### 4.2.1 SVM Performance and Decision Boundaries

The first benchmark model used was an SVM using an RBF kernel with a stratified 70/30 window split performed randomly across all the windows. This split was chosen as a starting point due to the number of training and test windows it gave, as well as maintaining the clean and attacked class proportions for both the training and test sets. Under the selected implementation setup summarized in Table 4, the SVM with an RBF kernel achieved an accuracy of 0.94 with AUC of approximately 0.99 on the GPS dataset, and an accuracy of 0.96 with AUC of approximately 0.93 on the Galileo dataset. These results indicate that the engineered feature space contains class-separating structure and that a nonlinear classifier can exploit that structure under the benchmark split. However, these values must be interpreted with caution because the random split allows windows from the same PRNs to appear in both training and test sets. This means the model is benchmarked in a regime where PRN-specific baselines may leak across the partition boundary. For this reason, the random split is treated only as a preliminary feature-discriminability benchmark, while the main generalization claims are based on the stricter validation protocols reported in the subsequent sections.

Error analysis indicates that the most classification errors were concentrated near the transition around 155 s, where the receiver was likely experiencing partial overlap between the authentic and replayed signals. Physically, this is exactly the region where feature patterns become ambiguous. The meaconing attack may not yet have fully dominated the tracking loops, and the receiver may still be reacting to a mixture of authentic and deceptive influences. The resulting windows occupy an intermediate region of the feature space that is neither fully clean nor fully attacked. This helps explain why the SVM produces false negatives in the early post-onset period. These errors are consistent with the transitional nature of replay takeover, where overlap between authentic and replayed signals produces ambiguous feature patterns.

The RBF kernel was used because the transition from overlap to takeover does not appear to follow a simple additive shift in the feature space. A linear model would assume

that attacked windows differ from normal windows mainly by moving in one fixed direction. The telemetry suggests otherwise. As attack begins to disturb the receiver, the interaction between power statistics, DLL stress, code variability, and doppler instability evolves in a nonlinear manner. The RBF kernel is therefore better suited to represent the curved decision boundaries needed to separate these classes. This benchmark is useful because it establishes a credible nonlinear baseline, against which the RF model can later be compared.

### 4.2.2 Random Forest: The Ensemble Advantage

RF had an accuracy of 0.9479 and AUC of 0.9904 for GPS, while for Galileo accuracy was 0.9887 with an AUC of 0.9966 under blocked chronological validation. RF was found to have a more consistent behaviour with Galileo where SVM had good recall but had a higher false alarm rate as shown in Table 4. This implies that RF could more easily model the nonlinear interactions between C/N0 variability, DLL instability, jitter features, and correlator distortion.

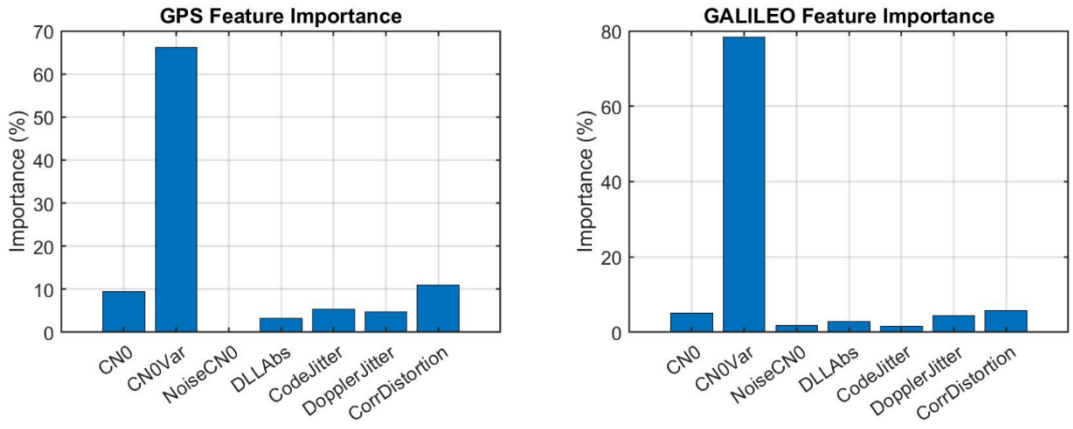


**Figure 7** Feature Importance Ranking.

Another advantage of the RF is that it allows us to calculate feature importances. The results in Figure 7 how different features contributed to the predictive capability of the detector for GPS and Galileo. In both cases, $C/N_o$ variance was the most important feature followed by Correlator Distortion. On the other hand, the least significant feature

was Noise $C/N_0$. We also observed in the ablation study that the Top 5 and Top 4 configurations were able to perform as well as the `Full7` feature configuration, while the `Min2` feature configuration shows a marginal drop in the AUC for the GPS data set as illustrated in Table 2. This suggests our model is not equally relying on all features, but mainly on a subset of features, which are somehow complementary.

**Table 2** Ablation Study comparing detection performance across different tracking-loop feature set sizes.

| Feature Set | GPS Accuracy | Galileo Accuracy | GPS AUC | Galileo AUC | GPS F1 | Galileo F1 |
|---|---|---|---|---|---|---|
| **Full7** | 0.94742 | 0.98872 | 0.99019 | 0.99683 | 0.96744 | 0.99318 |
| **Top5** | 0.95139 | 0.98872 | 0.98894 | 0.99372 | 0.96998 | 0.99318 |
| **Top4** | 0.94643 | 0.98872 | 0.98707 | 0.99319 | 0.96681 | 0.99318 |
| **Min2** | 0.95486 | 0.98872 | 0.98131 | 0.99323 | 0.97232 | 0.99318 |

## 4.3 Blocked Chronological Validation

Although the benchmark results in Table 2 show that the engineered feature space contains strong class-separating structure, the primary evaluation of the final model was performed using blocked chronological validation. In this protocol, training, validation, and testing were assigned to separate, non-overlapping temporal segments of the recording. This provides a stricter assessment than random partitioning because it preserves temporal order and reduces leakage between neighbouring windows. The blocked chronological split design and corresponding RF results are presented in the following subsections.

### 4.3.1 Blocked Chronological Split Design

The blocked chronological split was devised by splitting the windowed feature sequence into clean and attacked temporal blocks, which were further divided into training, validation and test partitions maintaining temporal continuity. Moreover, a buffer was also

preserved after the estimated attack onset to avoid training with the most transitional and confused windows in attacked data. This approach offers a more temporally controlled assessment than the random split. The sizes of the partitions for GPS and Galileo are shown in Table 3.

**Table 3** Blocked Chronological Split Summary.

| Constellation | Train Samples | Train Clean | Train Attack | Validation Samples | Validation Clean | Validation Attack | Test Samples | Test Clean | Test Attack |
|---|---|---|---|---|---|---|---|---|---|
| **GPS** | 3234 | 1400 | 1834 | 1260 | 420 | 840 | 2016 | 336 | 1680 |
| **Galileo** | 1848 | 800 | 1048 | 720 | 240 | 480 | 1152 | 192 | 960 |

### 4.3.2 Random Forest Performance Under Blocked Chronological Validation

The RF maintained its strong detection accuracy under blocked chronological validation, for both constellations. On GPS, the accuracy and AUC were 0.9479 and 0.9904, respectively. On the Galileo data, the equivalents were 0.9887 and 0.9966, respectively. These values are tabulated in Table 4 and further indicate that the detector passed this test where training, validation, and testing is done on non-overlapping temporal blocks. The significance of this more rigorous validation procedure is discussed in Chapter 5.

### 4.3.3 Learning Curve and Overfitting Analysis

The learning curve analysis of GPS shown in Figure 8 plots training and cross-validation accuracy as function of the amount of training windows. As the data sample size along the x axis grows, the accuracy of the training and the cross-validation converge smoothly and converge at a high level of performance. This is a critical convergence for our model; it indicates that the RF is learning the statistical edges of the meaconing transition, rather than simply trying to memorise high-frequency, localised multipath signatures found in

the training window. While the analysis for Galileo suggests that the accuracy for both training and validation stay at 1 across all sample sizes, which means that the model is classifying the validation set perfectly at every step in the experiment.

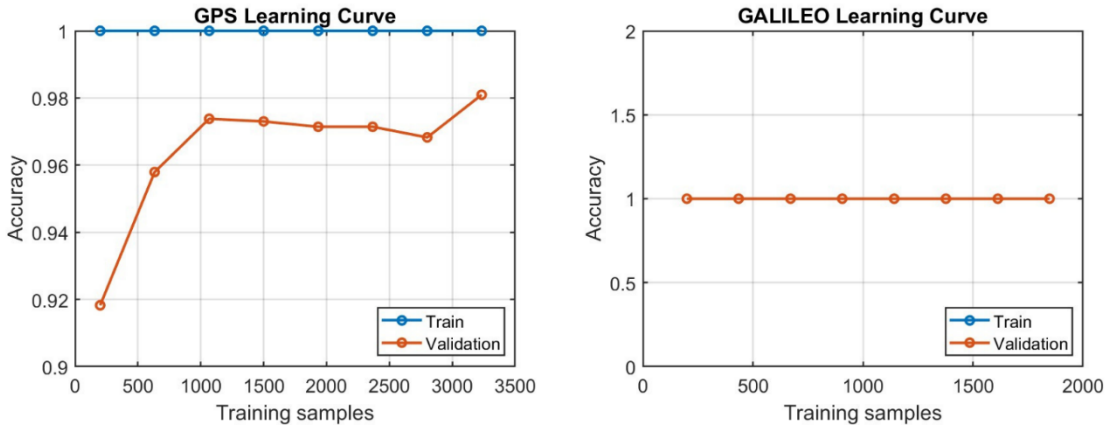


**Figure 8** Learning Curve Analysis.

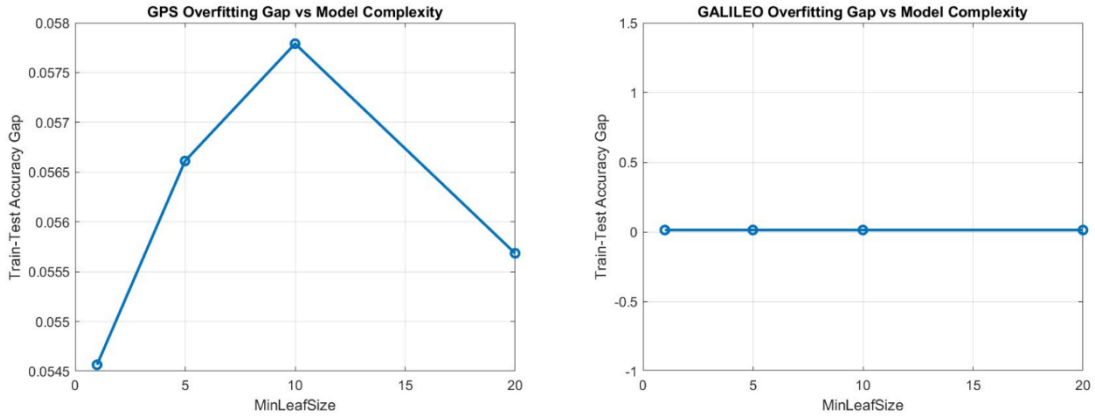


**Figure 9** Overfitting Gap vs Model Complexity.

Figure 9 shows the train-test accuracy gap as a function of the minimum leaf size, which measures the complexity of the model. The gap for GPS increases with model complexity up to a leaf size of 10 and then decreases at 20, indicating that model complexity influences generalization, but not in a strictly monotonic manner. For Galileo, the gap remains almost flat and near zero, suggesting that the model is insensitive to this hyperparameter.

Both models lack severe overfitting but there is a clear difference between them: GPS is sensitive to the leaf size hyperparameter (albeit in a minor way), but Galileo is not.

### 4.3.4 Confusion Matrix and ROC Analysis

The confusion matrices offer a concise representation of the classification behaviour under blocked chronological validation, as shown in Figure 10. The windows in these matrices are either clean or attacked, with class 0 indicating clean windows and class 1 indicating attacked windows. The diagonal entries are correct classifications, and the off-diagonal entries are errors in classifications. In GPS, the model was able to correctly classify 335 clean windows and 1570 attacked windows, while missing 110 attacked windows and false alarming in 1 window. In the case of Galileo, it was able to classify 192 clean windows and 945 attacked windows, with no false alarms and 15 missed attacked windows. The most distinguishing feature of both GPS and Galileo is the very small number of false alarms combined with a small number of missed attacked windows. This suggests that the detector was highly specific, with the little remaining error in false negatives instead of false positives.

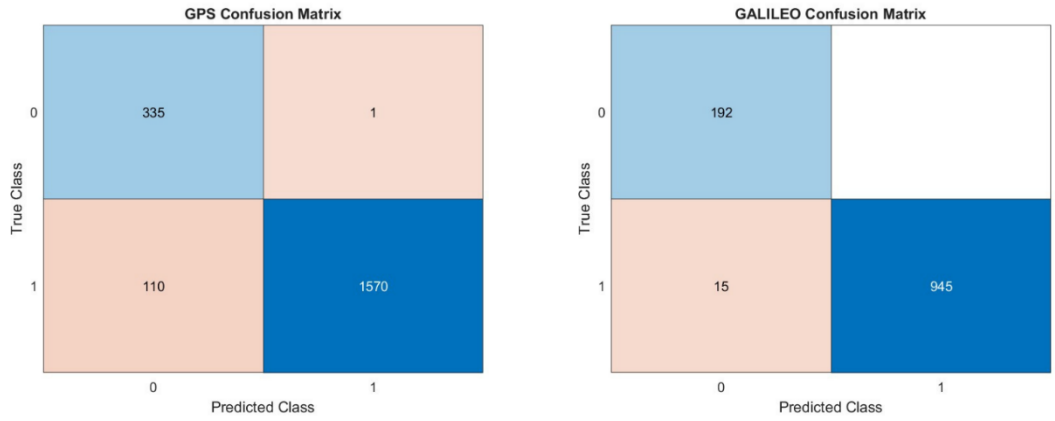


**Figure 10** Confusion Matrix.

The receiver operating characteristic (ROC) curves in Figure 11 compare the true positive rate to the false positive rate for all possible decision thresholds. Both constellations' curves follow a path that comes close to the upper left corner of the graph, with an AUC close to 0.99 for both constellations. This shows that the model's probability rankings

are able to discriminate between clean and attacked windows. Consequently, the system is not reliant on a rigid binary threshold, allowing system integrators to dynamically adjust the operating point based on specific mission risk tolerances.

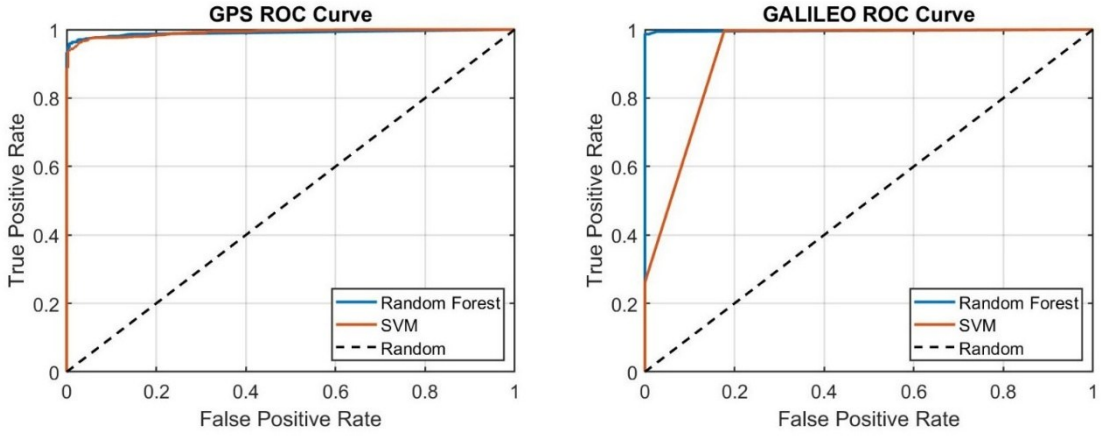


**Figure 11** ROC Curve.

**Table 4** Final Performance Metrics of the Implemented Meaconing Detection Models for GPS and Galileo.

| Constellation | Model | Accuracy | Precision | Recall | F1 Score | FAR | AUC | Mean TTD (sec) |
|---|---|---|---|---|---|---|---|---|
| **GPS** | RF | 0.9479 | 0.9993 | 0.9381 | 0.9677 | 0.0029 | 0.9904 | 1.47 |
| | SVM | 0.9494 | 0.9981 | 0.9410 | 0.9687 | 0.0080 | 0.9920 | 1.47 |
| **Galileo** | RF | 0.9887 | 1.0000 | 0.9864 | 0.9931 | 0.0000 | 0.9966 | 1 |
| | SVM | 0.9678 | 0.9656 | 0.9968 | 0.9810 | 0.1770 | 0.9335 | 1 |

## 4.4 Multiple PRN Split Generalization Results

The model was also tested on 20 independent multiple PRN splits on the combined GPS and Galileo data. This validation was added to measure satellite-independent performance in more than one fixed PRN-disjoint partition. Each time the model was developed, 70 percent of the available PRNs were used in developing the model, and the

remaining 30 percent were held out as an unseen PRN to test the model. This process gives a more rigorous and reliable estimate of the generalization since results reported are averaged across a variety of different combinations of unseen satellites and not based on a single selected PRN split.

The results of the split-wise test are presented in Table 5. The model had a mean accuracy of 0.9333 with a standard deviation of 0.0591, and a mean F1-score of 0.9462 with a standard deviation of 0.0516. The average of the AUCs was 0.9905 and the standard deviation was 0.0052 which shows that the classifier had a high average ranking ability both in clean and attacked windows with varying unseen PRN populations. It was also observed that the level of accuracy was always high with mean value of 0.9902 and a standard deviation of 0.0039, the false alarm rate was low with a mean value of 0.0192 and standard deviation of 0.0089. This implies that the detector did not frequently identify clean windows as being attacked across the recurrent PRN splits. The highest variability was witnessed in recall that had a mean of 0.9104 and a standard deviation of 0.0905. This indicates that sensitivity to detection was more so depending on the particular PRNs that were withheld to be tested. That is, although the model, in general, maintained good separability between clean and attacked windows, there were some unseen PRN combinations that were more difficult than others. This is expected in GNSS tracking data because individual satellites can differ in signal strength, elevation, multipath exposure, and nominal tracking-loop stability.

**Table 5** Multiple PRN Split Generalization Results Across 20 Splits.

| **Split** | **Accuracy** | **F1 Score** | **AUC** |
|---|---|---|---|
| **1** | 0.8074 | 0.8346 | 0.9804 |
| **2** | 0.9808 | 0.9857 | 0.9966 |
| **3** | 0.9817 | 0.9865 | 0.9959 |
| **4** | 0.9502 | 0.9620 | 0.9917 |
| **5** | 0.7926 | 0.8197 | 0.9756 |
| **6** | 0.9377 | 0.9523 | 0.9893 |

| | | | |
|---|---|---|---|
| **7** | 0.9889 | 0.9918 | 0.9960 |
| **8** | 0.8245 | 0.8515 | 0.9917 |
| **9** | 0.9465 | 0.9591 | 0.9910 |
| **10** | 0.9870 | 0.9905 | 0.9965 |
| **11** | 0.9211 | 0.9381 | 0.9870 |
| **12** | 0.9875 | 0.9908 | 0.9951 |
| **13** | 0.9331 | 0.9482 | 0.9890 |
| **14** | 0.9736 | 0.9804 | 0.9932 |
| **15** | 0.9234 | 0.9400 | 0.9891 |
| **16** | 0.9454 | 0.9581 | 0.9905 |
| **17** | 0.9104 | 0.9294 | 0.9877 |
| **18** | 0.9720 | 0.9791 | 0.9922 |
| **19** | 0.9625 | 0.9718 | 0.9924 |
| **20** | 0.9396 | 0.9536 | 0.9895 |

The results in Table 5 indicate that certain PRN combinations were more difficult than other ones. The lowest accuracy values were around 0.79 to 0.82, with some splits having accuracy values that were above 0.98. This difference proves that the performance cannot be reported by using a single PRN partition. Rather, mean, standard deviation, and confidence interval (CI) offer a more realistic estimate of generalization that is independent of satellites. Table 6 is a summary of the mean, standard deviation and 95% confidence intervals of the 20 multiple PRN splits. The student t -distribution was used in calculating the CI because the number of repeated PRN splits was limited to 20. For each metric, the confidence interval was computed as:

$$CI_{95} = \bar{x} \pm t_{0.975,K-1}\frac{s}{\sqrt{K}} \tag{15}$$

where $\bar{x}$ is the mean metric value across the repeated PRN splits, $s$ is the standard deviation, and $K$ is the number of splits. In this study, $K = 20$, so the t-critical value was taken with 19 degrees of freedom. The CI of the AUC was narrow (between 0.9881 and 0.9929), so the ranking ability of the classifier did not change significantly when various unseen PRN groups were used. Recall, on the other hand, had a broader confidence

interval, ranging between 0.8680 and 0.9528, showing that the sensitivity to detection varied more significantly, depending on which subset of withheld PRNs.

**Table 6** Multiple PRN Split Generalization Average Results.

| **Metric** | **Mean** | **Standard Deviation** | **95% CI Lower** | **95% CI Upper** |
|---|---|---|---|---|
| Accuracy | 0.9333 | 0.0591 | 0.9056 | 0.9610 |
| Precision | 0.9902 | 0.0039 | 0.9884 | 0.9920 |
| Recall | 0.9104 | 0.0905 | 0.8680 | 0.9528 |
| F1-score | 0.9462 | 0.0516 | 0.9221 | 0.9704 |
| FAR | 0.0192 | 0.0089 | 0.0150 | 0.0234 |
| AUC | 0.9905 | 0.0052 | 0.9881 | 0.9929 |

.

# 5 Discussion

The main insights of this thesis emerge from deployment-oriented evaluation protocols, particularly blocked chronological and cross satellite validation. The benchmark performance is severely compromised in the case of PRN-disjoint testing, where the test set PRNs do not appear at all in the training set, if the raw absolute features are used. The accuracy score drops to about 52% and the AUC to about 0.67. This indicates that the performance with random splits was at least partially due to leakage of the baseline information of each satellite.

## 5.1 The Generalization Gap and the Collapse of Absolute Features

The overall results suggest that the model learned not only replay-related behaviour but also PRN-specific operating signatures when absolute features were used. When that was taken away, the generalization performance degraded. In practice GNSS receivers see a dynamic set of satellites due to geometry, time of day, masking and motion. A detector that only performs well when tested on same PRNs that were used during training cannot be considered reliable for satellite independent deployment (J. Li et al., 2025). Thus, the cross-PRN failure shows that detector features cannot be used as satellite agnostic descriptors of attack, they are satellite dependent observations which are subject to domain shift (Siemuri et al., 2021). The detector must learn relative values with respect to local normal observations. This is why this thesis emphasises leakage-oriented performance over benchmarking. This is also supported by the blocked chronological validation results. The use of contiguous temporal blocks for training, validation and testing precluded random shuffling of adjacent windows across the boundaries. The fact that the detector retained good performance under this condition suggests it learned a persistent pattern in the meaconing signal rather than temporal structure in the vicinity of the transition points. Blocked chronological validation thus complements the PRN-disjoint analysis by reducing temporal leakage while preserving the sequential structure of the recording.

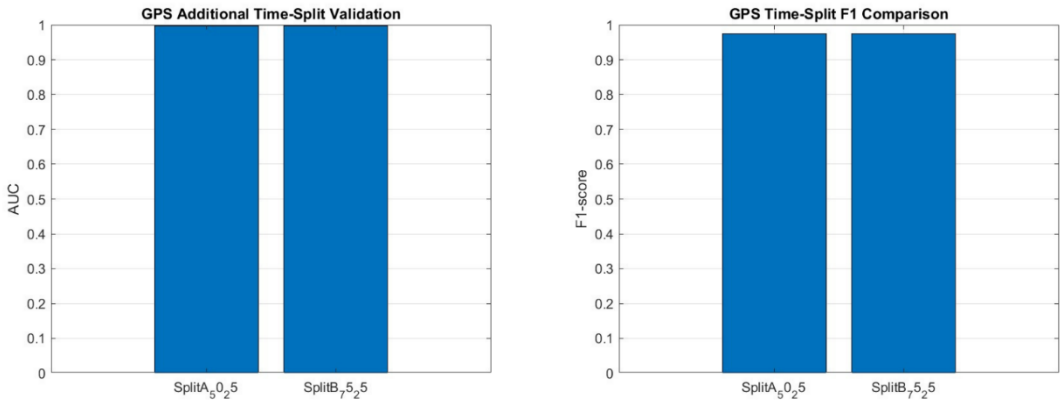


**Figure 12** GPS Time Split Validation.

To determine whether the model performance is dependent on a particular temporal portion of the data, time-split validation experiments were also conducted. The following splits were tested: Split A (50-25) where the model is trained on the first 50% of the time-series and evaluated on the next 25%, and Split B (75-25) where the training period is extended (75%) and the model is evaluated on the remaining segment. Both AUC and the F1-score were computed on the splits for the GPS and Galileo constellations shown in Figures 12 and 13.

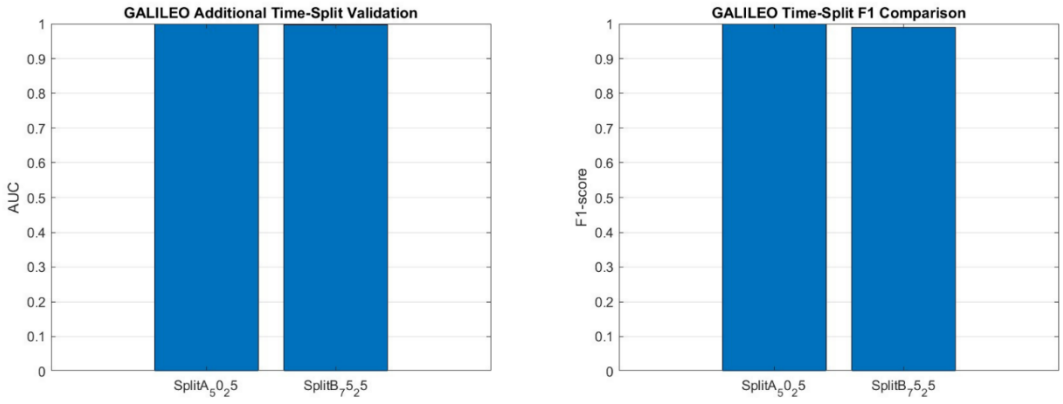


**Figure 13** Galileo Time Split Validation.

Both metrics were consistently higher than 0.95 for all considered splits for the two constellations, suggesting that the model yields consistent detection performance across

different temporal splits. This suggests that the meaconing patterns are not only detected within a specific time window but are also representative of different time windows of the recording. This also confirms the robustness of the proposed detection approach in different scenarios.

## 5.2 Solving the Generalization Gap Through PRN Baseline Normalization

This thesis assesses the PRN baseline normalization as a feasible means to decrease the satellite-specific baseline bias by transforming each feature into a deviation of the pre-attack PRN baseline. This subtraction transforms the learning problem. Rather than enquiring whether the magnitude of a raw feature is globally suspicious, it enquires whether the current magnitude is unusual for that satellite. When we reframe the problem in this manner, the results improve sharply. After normalizing and cross-evaluating, the AUC = 0.9904 and accuracy = 94.79% with very high precision and specificity. This is the best evidence that replay is a change from baseline effect.

This result reveals more information about attack signatures rather than the random-split benchmark, because it tests whether the attack signature remains detectable after reducing PRN identity cues. It demonstrates the discriminative structure remains without PRN identity cues. In fact, it becomes clearer. The classifier does not need to memorise which satellite is usually strong, weak, noisy or smooth. It merely needs to detect when a channel starts to act differently. This is more in line with how a realistic receiver would check integrity. It also sheds light on why it performs better: slow changes in geometry or link budget will naturally shift the absolute operating point (Chen et al., 2022), but attack onset still appears as a local statistical innovation. For this reason, PRN-relative normalization is more than a preprocessing convenience. Practical advantage of this normalization is its ease of implementation. The baseline is only computed from clean windows in the past, so the same logic can be applied online (without future knowledge). This is a crucial point because many normalization techniques work offline but cannot be implemented in a real-time receiver. The testing process and the implementation are in sync. This allows the thesis to be more engineering-focused: the lab fix becomes the real-time fix.

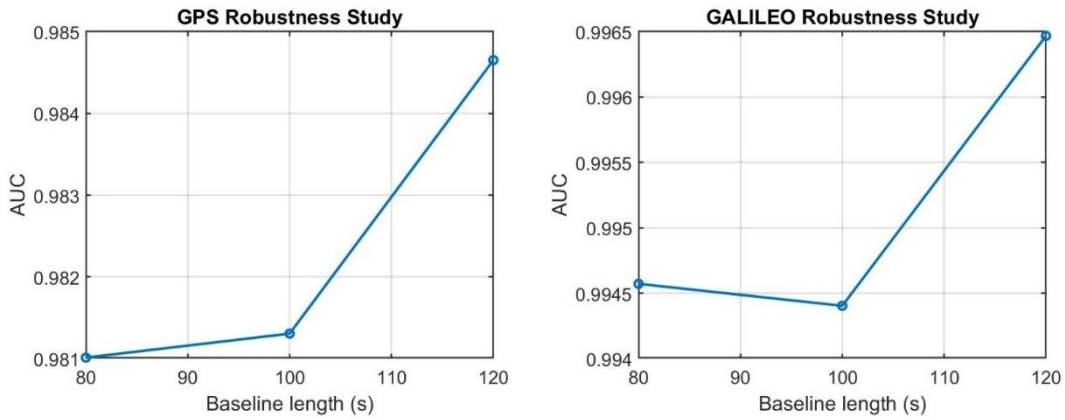


**Figure 14** Robustness Study.

The plot in Figure 14 shows that when the length of the baseline estimation window is varied between 80, 100 and 120 seconds, the cross-PRN AUC retains its shape close to 1.0. This verifies that the PRN baseline normalization is able to extract relative attack behaviour and is very resilient with respect to absolute power differences between high-elevation and low-elevation satellites. It also indicates that an unreasonably long "clean" initialisation period to build a perimeter is not required. The model continues to place attacked windows ahead of normal windows, even with a reduced baseline window. The small drop in the AUC of GPS is more likely related to a calibration offset than to a change in separability.

## 5.3 Multiple PRN Split Validation

The multiple PRN split validation gives a more realistic picture of satellite-independent performance than a single PRN-disjoint split or LOPO validation alone. Since the model was evaluated across 20 repeated 70/30 PRN splits, the reported results are not dependent on one favourable or unfavourable satellite partition. The mean AUC of 0.9905, with a narrow 95% confidence interval from 0.9881 to 0.9929, indicates that the model maintained stable ranking ability across different unseen PRN groups. In most split combinations, attacked windows were assigned higher risk scores than clean windows, which

supports the view that the extracted tracking-loop features captured replay-related behaviour rather than only memorizing PRN-specific baselines.

However, the results also show that generalization was not perfectly uniform across all PRN combinations. Recall had a mean value of 0.9104, a standard deviation of 0.0905, and a wider 95% confidence interval from 0.8680 to 0.9528. This indicates that some withheld PRN groups contained attack windows that were more difficult to detect. This variability is expected in GNSS tracking data, because individual satellites can differ in signal strength, elevation, Doppler behaviour, tracking noise, and multipath exposure. Even after PRN-relative normalization, some PRNs may still show weaker or less distinct attack-induced deviations than others.

The difference between stable AUC and more variable recall is important. AUC reflects the model's ability to rank clean and attacked windows, while recall depends on the selected decision threshold. Therefore, the stable AUC but wider recall interval suggests that the feature space remained informative across unseen PRNs, but a fixed threshold was not equally effective for all satellite subsets. This supports the use of score-based decision policies rather than relying only on a rigid binary alarm threshold.

Overall, the multiple PRN split validation strengthens the thesis argument by showing both the capability and the limitation of the proposed method. The model achieved strong average discrimination across unseen PRN groups, but the variation in recall confirms that satellite-dependent effects were not fully eliminated. Therefore, the results support satellite-independent detection under the evaluated dataset, while also showing that further validation across additional recordings, receiver conditions, and PRN geometries is needed before broader operational claims can be made

## 5.4 Time-to-Detection

Better classification is not enough on its own. A receiver security mechanism must also be fast enough to be effective before replay induced bias affects time or navigation decisions, which is vital to avoid catastrophic consequences in autonomous UAV (Manesh et al., 2019), and power grid phase synchronization (Siamak et al., 2021). For this reason, TTD was measured with the best possible scenario: LOPO validation, PRN normalization,

the top 5 feature subset, a 200-tree RF and a stability rule requiring K = 3 consecutive positive windows. With this setup, TTD is from attackStartSec to the end of the third consecutive positive window. The K = 3 stability rule is important because it demonstrates that the detector is not responding to single-window changes. The K = 3 rule provides a temporal consistency filter to eliminate nuisance alarms from single spikes. That confirmed detection is still accomplished rapidly for most of the PRNs tested under this stricter rule. This implies that the technique can still be responsive and less sensitive to the changes of single windows. It can thus be helpful in safety-of-life autonomy (Manesh et al., 2019) as an early warning layer but must be further validated before it can be used.

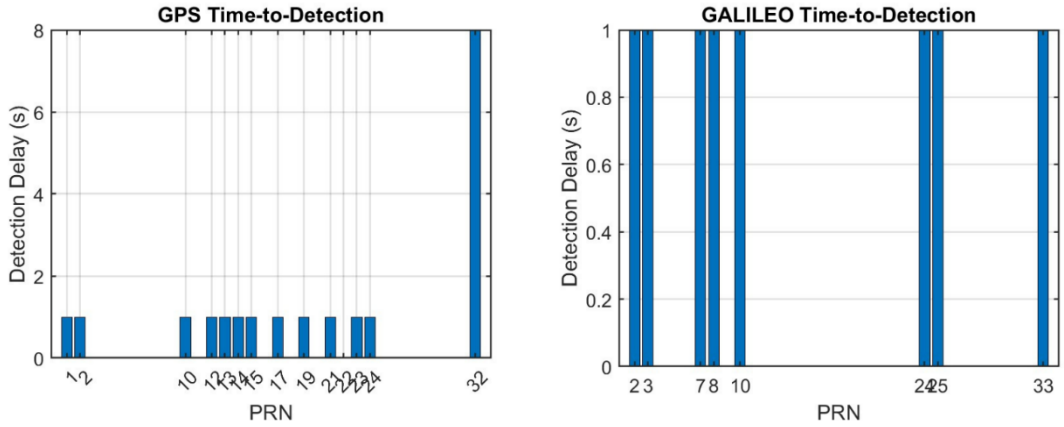


**Figure 15** Time-to-Detection.

TTD plots in Figures 15 show the detection latency obtained under the final stability rule. Despite enforcing K = 3 consecutive positive windows to suppress transient RF noise, most tracked satellites still trigger confirmed detection within short delay intervals. This indicates that the proposed method retains useful responsiveness even after temporal consistency is imposed.

## 5.5 The Edge Case of PRN 32 and the Need for Score-Based Decision Policy

PRN 32 is the most illuminating outlier in the thesis as it illustrates the distinction between the quality of ranking and policy of decision making. The AUC for this PRN under LOPO evaluation was approximately 0.9607, which means that the classifier distinguished between the attacked and normal windows fairly well. But at the default probability cut-off (0.5), the recall was zero. This might seem at first glance counter-intuitive, but in fact the scores were informative and simply below the default threshold. By adjusting the threshold to about 0.65, the performance improved dramatically, regaining about 89.6% accuracy and 90.8% recall while leaving the AUC unchanged. So the problem was not inherent to the features, but rather incorrect thresholding. This is in line with the audit of nominal heterogeneity. PRN 32 was already known to be a noisy channel, probably due to low elevation and other geometric factors. In these channels, the distribution of predicted scores might be squashed so that attack windows are suspicious according to normal windows, but not suspicious enough to pass a universal threshold. This is precisely why AUC matters so much in cross-PRN analysis. Threshold-independent ranking can remain strong even when one threshold performs badly. The PRN 32 case therefore strengthens the thesis argument for exporting a continuous spoofing-likelihood score rather than relying on a rigid binary alarm, facilitating better integration with downstream sensor fusion engines (Dang et al., 2022).

## 5.6 Computational Complexity and Embedded Feasibility

A final discussion point is whether the proposed detector is practically feasible for autonomous and embedded systems. The results suggest that such deployment appears feasible in principle, subject to receiver-specific implementation and benchmarking. Feature extraction operates once per PRN, once per second over 1 second windows of 1000 samples. The required operations are simple window statistics such as means, variances, and dispersion measures, all with linear complexity in the number of samples and small constant factors. Baseline normalization is even cheaper, requiring only subtraction of a

stored mean vector from the current feature vector. RF inference with 200 trees also remains feasible because it occurs at a 1 Hz decision rate rather than at the raw sample rate. In practical terms, this suggests that the added security logic is lightweight relative to the core receiver tracking workload.

Memory requirements are similarly modest. A real-time implementation would need to store rolling statistics or a short window buffer, a baseline vector for each PRN, and a short history for the K = 3 stability logic. This is important because many promising detection schemes cannot be deployed because they are too big, not interpretable, or external sensors are required. In contrast, the current pipeline is small, transparent and deterministic. These features are desirable in safety-critical applications where small, deterministic and interpretable systems are preferred. In terms of positioning, the technique should be seen as part of a larger integrity architecture rather than a silver bullet. The attractive feature is that it leverages on receiver-internal telemetry that is already present in many software-defined and some embedded receiver systems. Meaconing can preserve the content of GNSS signals while corrupting timing and/or navigation integrity through delayed replay (K. Zhang et al., 2022), internal telemetry monitoring offers a particularly valuable layer of protection. A lightweight, fast tracking-loop-based score might assist upper-level integrity management operations like measurement weighting, channel removal or estimator protection. In this regard, the method has impact beyond the integrity of the classifier itself to system integrity. Minimising the size of the feature space while ensuring detection integrity is essential to ensure real-time operation (Zarrinnegar et al., 2025).

The overfitting diagnostics also show the effectiveness of the final configuration. First, the learning curve in Figure 8 showed that as the amount of training data increased, the training and validation gap converged and did not diverge. The model-complexity analysis in Figure 9 also showed that the chosen configuration with minimum leaf of 10 is an appropriate configuration for the model to have the right balance between complexity and overfitting. The feature-importance in Figure 7 and the ablation test in Table 2 provide some information about how meaconing is captured in the analysed data sets. In both constellations, the most important feature is $C/N_o$ variance and Correlator

Distortion, while Noise $C/N_o$ is the least significant feature. It seems that, while the detector does not require a great number of features, it requires a number of different features. In terms of model design, an efficient detector is thus not necessarily the detector using the largest number of features, but the detector which efficiently represents the main physical mechanisms of the attack. This is important in practice because this suggests that it is possible to use relatively small models

# 6 Conclusion and Future Work

## 6.1 Summary of Findings

The thesis developed and evaluated a machine-learning approach to detect GNSS meaconing attacks with the aid of receiver-internal tracking-loop telemetry. Data was processed using the FGI-GSRx software receiver to perform validation on the GPS L1 and Galileo E1 constellations. The experiments suggest that under the evaluated dataset and validation protocols, meaconing produces a multi-dimensional statistical pattern in the tracking-loop domain, resulting in distinguishable clean and attacked windows. A random forest classifier (with 200 trees and minimum leaf size 10) was able to achieve good detection performance, with the AUC of 0.9904 for GPS and 0.9966 for Galileo. Also, the multiple PRN split validation over combined GPS and Galileo PRN was found to have an average AUC of 0.9905 ± 0.0052 and F1-score of 0.9462 ± 0.0516, showing that the model had strong overall average discrimination ability across unseen subsets of satellites. A comprehensive feature importance analysis showed that the features' predictive significance was not the same for both constellations. In both GPS and Galileo, $C/N_0$ variance was the top feature, while Correlator Distortion was also among the top features. On the other hand, noise $C/N_0$ was the weakest. These results suggest that, for the considered configurations, meaconing is best captured by a combination of power variability and correlator distortion rather than through a balanced contribution of all features. The ablation study also revealed a reduced Top 5 feature set was sufficient to achieve excellent classification while decreasing the feature space.

The overall results of the study indicate that the significance of the thesis is not restricted to proving that ML can be utilized to classify meaconing attacks from GNSS receivers using the telemetry data received by the receivers. It primarily contributes to the methodological design of the detection pipeline, specifically by generating detailed interpretable tracking-loop features, labelling the onset of attacks through data-driven methods, normalizing the PRN with respect to its own individual baseline, and a deployment-oriented validation. First, the results demonstrate that random window splits can

significantly inflate results by enabling PRN-specific baseline information to inform the evaluation. Second, they demonstrate that absolute tracking features are not good predictors for satellite-independent attack descriptors. Third, they show that simple causal PRN-relative normalization does restore the cross-satellite generalization ability and produces a detector that is fast, compact and accurate.

## 6.2 Theoretical and Methodological Contributions

The thesis contribution is that it identifies PRN-dependent baseline effects and reduces their influence through PRN-relative feature normalization and stricter validation protocols. This thesis also demonstrates that conventional cross-validation with random splitting can be susceptible to leakage, enabling models to learn absolute operating points for satellites, rather than the typical relative signatures of attacks. Another methodological contribution is the use of blocked chronological validation and PRN-disjoint analysis, which prevent both temporal leakage and PRN-specific leakage in the analysis. This assessment is further strengthened through various PRN split validation, where the model is repeatedly trained and tested on different unknown satellite groups and performance can be reported as a mean ± standard deviation, rather than as a single split result. Stiff PRN-disjoint testing reduced the unnormalized baseline's accuracy to 52%, demonstrating that absolute tracking magnitudes are satellite-specific observations, rather than reliable indicators of security vulnerabilities. To address this, the thesis proposed Local PRN-Relative Normalization, a causal normalization approach that converts raw telemetry values into relative deviations from a satellite's co-localized pre-attack behaviour. This approach allowed the separation of satellite identity from attack and also enhances the cross-satellite generalization under the experimental conditions. This interpretation is supported by the similar ROC behaviour seen with GPS and Galileo though more testing on other datasets is necessary to confirm the interpretation.

## 6.3 Practical Implications

The proposed model may be relevant to GNSS-dependent systems that rely on reliable timing or positioning, such as telecommunications, transport, power-grid synchronization, and autonomous systems. In this context the model can be used as an early warning layer that detects unusual tracking-loop behaviour, but it is not a standalone solution. In general, it is part of a larger GNSS security framework.

**Embedded Feasibility:** The use of common tracking-loop statistics (means, variances and jitter) and the Top 5 feature set achieves $O(N)$ computational complexity. At a 1 Hz alerting rate, RF scoring is light on processing and memory.

**Operational Latency (TTD):** The framework showed low TTD ranges for 93% of the constellation with a hard stability rule of K=3 to eliminate transient RF noise. For autonomous UAV, this real-time alerting is enough to trigger "Hold Position" or inertial-fallback protocols before physical hijacking. For high-voltage power systems, this enables PMUs to immediately reject tampered time-of-arrival data, thus avoiding fatal phase-angle estimation errors.

**Score-Based Decision Policies:** The analysis of low-elevation edge case (e.g., PRN 32) demonstrated the need to provide a deception probability score rather than a hard binary alarm. This enables a more flexible choice of thresholds with respect to mission risk and signal environment.

## 6.4 Limitations and Future Work

The results are to be read as an indication of practicability under the FGI-SpoofRepo Meaconing DFMC dataset instead of being considered as operational validation across all GNSS receivers and conditions. The dataset is short in time, receiver is stationary and the assessment is done based on a particular software-defined receiver processing chain. These limitations restrict the applicability of the reported performance to dynamic receivers, various antenna scenarios, different levels of replay power, or commercial receiver architectures. While the framework demonstrates high resilience and accuracy, certain limitations present opportunities for continued research:

**Dynamic Receiver Environments:** The dataset primarily reflects a static or semi-static receiver environment. High-velocity receivers (e.g., fixed-wing UAVs) experience rapid natural fluctuations in doppler and multipath fading. Future work must evaluate the robustness of the 120-second baseline normalization under highly dynamic conditions, potentially exploring adaptive baselines that gradually reduce the influence of clean data while tracking recent nominal behaviour.

**Receiver Architecture Dependency:** The telemetry utilized in this study was obtained by processing raw IQ data with the FGI-GSRx software-defined receiver in MATLAB. Subsequent research should investigate mapping these ML algorithms directly onto commercial hardware receivers. Another practical limitation is software accessibility. Since MATLAB based experiments may depend on licensed toolboxes, future work should also explore similar pipeline implementation using open-source or open-access platforms (e.g. Python).

**Integration with Cryptographic Authentication:** Meaconing's defining threat is its ability to bypass message-level cryptography by replaying authentic data. A highly promising avenue for future work is the fusion of this tracking-loop ML framework with open service navigation message authentication (OSNMA). OSNMA is an open-service authentication mechanism intended for civilian and public users. It allows compatible receivers to verify that received Galileo navigation message is authentic and has not been modified. While OSNMA protects against synthesized spoofing, it remains theoretically vulnerable to distance-decreasing (DD) replay attacks. Therefore, tracking-loop instability scores can be used as a secondary verification layer for OSNMA-authenticated signals.

**Swarm and Collaborative Defense:** The PRN-relative normalization logic can be expanded to multi-receiver or networked GNSS monitoring in the future. When multiple receivers detect a common GNSS environment, the tracking loop features could be compared to determine spatial diversity. This cross-receiver feature corelation can be used to lower false alarms in urban dense multipath environments and increase the resilience to meaconing attack in autonomous or networked systems. This would need more datasets with synchronized receiver and is therefore not within the scope of this thesis as it is only single receiver.

# References

Ahmed, W., Masood, A., Manzoor, J., & Sedat Akleylek. (2025). Automatic de-pendent surveillance-broadcast (ADS-B) anomalous messages and attack type detection: deep learning-based architecture. *PeerJ Computer Science*, 11, e2886–e2886. https://doi.org/10.7717/peerj-cs.2886

Akos, D. M. (2012). Who's afraid of the spoofer? GPS/GNSS spoofing Detection via Automatic Gain Control (AGC). *NAVIGATION Journal of the Institute of Navigation*, *59*(4), 281–290. https://doi.org/10.1002/navi.19

Babić, K., Balić, M. and Begušić, D. (2025). GNSS Spoofing Detection Based on Wavelets and Machine Learning. *Electronics*, 14(12),2391. https://doi.org/10.3390/electronics14122391

Broumandan, A., Kennedy, S., & Schleppe, J. (2020). Demonstration of a Multi-Layer Spoofing Detection Implemented in a High Precision GNSS Receiver [Conference paper]. *2020 IEEE/ION Position, Location and Navigation Symposium (PLANS)*, 538–547. https://doi.org/10.1109/PLANS46316.2020.9109842

Bull, T. (2010). A new high performance way of detecting and mitigating the Jamming Meaconing and spoofing of commercial GNSS signals [Conference paper]. *2010 5th ESA Workshop on Satellite Navigation Technologies and European Workshop on GNSS Signals and Signal Processing (NAVITEC)*, 1–5. https://doi.org/10.1109/NAVITEC.2010.5708050

Chen, Z., Li, J., Li, J., Zhu, X., & Li, C. (2022). GNSS Multiparameter Spoofing Detection Method Based on Support Vector Machine. *IEEE Sensors Journal*, *22*(18), 17864–17874. https://doi.org/10.1109/JSEN.2022.3193388

Dang, Y., Benzaïd, C., Yang, B., Taleb, T., & Shen, Y. (2022). Deep-Ensemble-Learning-Based GPS Spoofing Detection for Cellular-Connected UAVs. *IEEE Internet of Things Journal*, *9*(24), 25068–25085. https://doi.org/10.1109/JIOT.2022.3195320

Eldosouky, A., Ferdowsi, A. and Saad, W. (2020). Drones in Distress: A Game-Theoretic Countermeasure for Protecting UAVs Against GPS Spoofing. *IEEE Internet of Things Journal*, 7(4), pp. 2840–2854. https://doi.org/10.1109/JIOT.2019.2963337

Ghizzo, E., Pena, A. G., Lesouple, J., Milner, C., & Macabiau, C. (2024). Assessing GNSS Carrier-to-Noise-Density Ratio Estimation in The Presence of Meaconer Interference [Conference paper]. *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 8971–8975. https://doi.org/10.1109/ICASSP48485.2024.10448170

Han, S., Chen, L., Meng, W., & Li, C. (2017). Improve the Security of GNSS Receivers Through Spoofing Mitigation. IEEE Access, 5, 21057–21069. https://doi.org/10.1109/access.2017.2754414

Iqbal, A., Muhammad Naveed Aman, & Biplab Sikdar. (2024). A Deep Learning based Induced GNSS Spoof Detection Framework. *IEEE Transactions on Machine Learning in Communications and Networking*, 2, 457–478. https://doi.org/10.1109/tmlcn.2024.3386649

Islam, S., Bhuiyan, M.Z.H., Liaquat, M., Pääkkönen, I. and Kaasalainen, S. (2024a). An open GNSS spoofing data repository: characterization and impact analysis with FGI-GSRx open-source software-defined receiver. *GPS Solutions*, 28(176). https://doi.org/10.1007/s10291-024-01719-2

Islam, S., Bhuiyan, M. Z. H., Liaquat, M., Pääkkönen, I., & Kaasalainen, S. (2024b). FGI's GNSS Spoofing Dataset Repository (FGI-SpoofRepo) (Version 2). National Land Survey of Finland, FGI Dept. of Navigation and positioning. https://doi.org/10.23729/7a648509-2ca8-4a7d-8223-0b429182f857

Jafarnia Jahromi, A., Broumandan, A., Nielsen, J., & Lachapelle, G. (2012a). GPS spoofer countermeasure effectiveness based on signal strength, noise power, and C/N0 measurements. *International Journal of Satellite Communications and Networking*, 30(4), 181–191. https://doi.org/10.1002/sat.1012

Jafarnia-Jahromi, A., Broumandan, A., Nielsen, J., & Lachapelle, G. (2012b). GPS Vulnerability to Spoofing Threats and a Review of Antispoofing Techniques. *International Journal of Navigation and Observation*, 2012, 1, 127072. https://doi.org/10.1155/2012/127072

Jiang, C., Chen, S., Zhang, B., Chen, Y., Bo, Y., & Feng, Z. (2018). Effective-ness Analysis of the Covariance Matrix for Spoofing Detection Application. *2018 Ubiquitous*

*Positioning, Indoor Navigation and Location-Based Services (UPINLBS)*, 1–5. https://doi.org/10.1109/upinlbs.2018.8559759

Kapoor, S., & Narayanan, A. (2023). Leakage and the reproducibility crisis in machine-learning-based science. *Patterns*, *4*(9), 100804. https://doi.org/10.1016/j.patter.2023.100804

Li, J., Chen, Z., Yuan, X., Xie, T., Xu, Y., Zheng, Z. and Zhu, X. (2025). A real-time GNSS time spoofing detection framework based on feature processing. *GPS Solutions*, 29(45). https://doi.org/10.1007/s10291-024-01802-8

Li, Y., Cai, C., & Xu, Z. (2022). A combined elevation angle and C/N0 weighting method for GNSS PPP on Xiaomi MI8 smartphones. *Sensors*, *22*(7), 2804. https://doi.org/10.3390/s22072804

Liu, S., Li, S., Zheng, J., Fu, Q., & Yuan, Y. (2020). C/N0 Estimator Based on the Adaptive Strong Tracking Kalman Filter for GNSS Vector Receivers. Sensors, 20(3), 739. https://doi.org/10.3390/s20030739

Manesh, M., Kenney, J., Hu, W.C., Devabhaktuni, V.K. and Kaabouch, N. (2019). Detection of GPS Spoofing Attacks on Unmanned Aerial Systems. *2019 16th IEEE Annual Consumer Communications & Networking Conference (CCNC)*. https://doi.org/10.1109/CCNC.2019.8651804

Marnach, D., Mauw, S., Martins, M. and Harpes, C. (2013). Detecting Meaconing Attacks by Analysing the Clock Bias of GNSS Receivers. *Artificial Satellites*, 48(2), pp. 63–83. https://doi.org/10.2478/arsa-2013-0006

Noman Chowdhury, A. A., Ahmadi, E., Elmusrati, M., Kuusniemi, H., & Boutellier, J. (2026). Reinforcement Learning for GNSS Spoofing Detection: A Multi-Class DQN Approach with TEXBAT. *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 19862–19866. https://doi.org/10.1109/icassp55912.2026.11462191

OpenAI. (2026). *ChatGPT (GPT-5.3) [Large language model]*. https://chat.openai.com/

Panice, G., Luongo, S., Gigante, G., Pascarella, D., di Benedetto, C., Vozella, A., & Pescapè, A. (2017). A SVM-based detection approach for GPS spoofing attacks to UAV

[Conference paper]. *2017 23rd International Conference on Automation and Computing (ICAC)*, 1–11. https://doi.org/10.23919/IConAC.2017.8081999

Psiaki, M. L., O'Hanlon, B. W., Bhatti, J. A., Shepard, D. P., & Humphreys, T. E. (2013). GPS Spoofing Detection via Dual-Receiver Correlation of Military Signals. *IEEE Transactions on Aerospace and Electronic Systems*, *49*(4), 2250–2267. https://doi.org/10.1109/TAES.2013.6621814

Psiaki, M. L., & Humphreys, T. E. (2016). GNSS Spoofing and Detection. *Proceedings of the IEEE*, *104*(6), 1258–1270. https://doi.org/10.1109/JPROC.2016.2526658

Paziewski, J., Sieradzki, R., & Baryla, R. (2019). Signal characterization and assessment of code GNSS positioning with low-power consumption smartphones. *GPS Solutions*, *23*(4). https://doi.org/10.1007/s10291-019-0892-5

Safi, M. (2025). *GNSS Timing Spoofing Detection: Methods and Analysis using Jammer test data*. Master's Thesis. University of Vaasa. https://osuva.uwasa.fi/handle/11111/12427

Seco-Granados, G., Gómez-Casco, D., López-Salcedo, J. A., & Fernández-Hernández, I. (2021). Detection of replay attacks to GNSS based on partial correlations and authentication data unpredictability. *GPS Solutions*, *25*(2), 33. https://doi.org/10.1007/s10291-020-01049-z

Semanjski, S., Semanjski, I., De Wilde, W. and Muls, A. (2020a). Use of Supervised Machine Learning for GNSS Signal Spoofing Detection with Validation on Real-World Meaconing and Spoofing Data—Part I. *Sensors*, 20(4), 1171. https://doi.org/10.3390/s20041171

Shafique, A., Mehmood, A. and Elhadef, M. (2021). Detecting Signal Spoofing Attack in UAVs Using Machine Learning Models. *IEEE Access*, 9, pp. 93803–93815. https://doi.org/10.1109/ACCESS.2021.3089847

Shang, X., Sun, F., Zhang, L., Cui, J. and Zhang, Y. (2022). Detection and mitigation of GNSS spoofing via the pseudorange difference between epochs in a multicorrelator receiver. *GPS Solutions*, 26(2), 37. https://doi.org/10.1007/s10291-022-01224-4

Siamak, S., Dehghani, M. and Mohammadi, M. (2021). Dynamic GPS Spoofing Attack Detection, Localization, and Measurement Correction Exploiting PMU and SCADA.

*IEEE Systems Journal*, 15(2), pp. 2531–2540. https://doi.org/10.1109/JSYST.2020.3001016

Siemuri, A., Kuusniemi, H., Elmusrati, M. S., Välisuo, P., & Shamsuzzoha, A. (2021). Machine Learning Utilization in GNSS—Use Cases, Challenges and Future Applications. *2021 International Conference on Localization and GNSS (ICL-GNSS)*, 1–6. https://doi.org/10.1109/ICL-GNSS51451.2021.9452295

Steiner, J., Pleninger, S., & Hospodka, J. (2024). Assessing the Vulnerability of Aviation Systems to GNSS Meaconing Attacks. *2024 New Trends in Civil Aviation (NTCA)*, 213–218. https://doi.org/10.23919/NTCA60572.2024.10517809

Sun, C., Cheong, J. W., Dempster, A. G., Zhao, H., & Feng, W. (2018). GNSS Spoofing Detection by Means of Signal Quality Monitoring (SQM) Metric Combinations. IEEE Access, 6, 66428–66441. https://doi.org/10.1109/access.2018.2875948

Tatbul, N., Lee, T. J., Zdonik, S., Alam, M., & Gottschlich, J. (2018). Precision and Recall for Time Series. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds), *Advances in Neural Information Processing Systems* (Vol. 31). Retrieved from https://proceedings.neurips.cc/paper_files/paper/2018/file/8f468c873a32bb0619eaeb2050ba45d1-Paper.pdf

Wei, X., & Sikdar, B. (2019). Impact of GPS Time Spoofing Attacks on Cyber Physical Systems. *2019 IEEE International Conference on Industrial Technology (ICIT)*, 1155–1160. https://doi.org/10.1109/ICIT.2019.8755016

Wührl, T., Baross, M. T., Gyányi, S., & Varga, P. J. (2023). 5G Synchronization Problems with GNSS Interference. 2023 IEEE 6th International Conference and Workshop Óbuda on Electrical and Power Engineering (CANDO-EPE), 000149–000154. https://doi.org/10.1109/cando-epe60507.2023.10417998

Yakkati, R. R., Pardhasaradhi, B., Zhou, J., & Cenkeramaddi, L. R. (2022). A Machine Learning based GNSS Signal Classification [Conference paper]. *2022 IEEE International Symposium on Smart Electronic Systems (iSES)*, 532–535. https://doi.org/10.1109/iSES54909.2022.00116

Zarrinnegar, K., Sormayli, J. and Mosavi, M.R. (2025). Robust GNSS spoofing detector using optimized machine learning model on embedded platforms. *GPS Solutions*, 30(1), 24. https://doi.org/10.1007/s10291-025-01985-8

Zhang, H., Peng, S., Liu, L., Su, S. and Cao, Y. (2020). Review on GPS spoofing-based time synchronisation attack on power system. *IET Generation, Transmission & Distribution*, 14(20), pp. 4301–4309. https://doi.org/10.1049/iet-gtd.2020.0253

Zhang, K., Larsson, E.G. and Papadimitratos, P. (2022). Protecting GNSS Open Service Navigation Message Authentication Against Distance-Decreasing Attacks. *IEEE Transactions on Aerospace and Electronic Systems*, 58(2), pp. 1224–1240. https://doi.org/10.1109/TAES.2021.3122512

Zhu, X., Lu, Z., Hua, T., Yang, F., Tu, G., & Chen, X. (2022). A novel GPS meaconing spoofing detection technique based on improved ratio combined with carrier-to-noise moving variance. *Electronics, 11*(5), Article 738. https://doi.org/10.3390/electronics11050738

Zuo, S., Liu, Y., Zhang, D., Xin, P., & Liu, T. (2021). Detection of GPS Spoofing Attacks Based on Isolation Forest. *2021 IEEE 9th International Conference on Information, Communication and Networks (ICICN)*, 357–361. https://doi.org/10.1109/icicn52636.2021.9673863

# Appendices

## **Appendix 1.** AI Prompt for Data Processing Pipeline Figure

Create a high-resolution academic infographic diagram (≥400 DPI, 1:1 aspect ratio) showing a GNSS data processing pipeline from raw signal to feature dataset. Include the following steps in order:

1. Raw GNSS I/Q Data (GPS L1 and Galileo E1)
2. Signal processing using MATLAB-based FGI-GSRx receiver (tracking loops: DLL, PLL, correlators)
3. Tracking loop outputs per epoch (CN0, DLL, Doppler, Code Phase, correlator outputs)
4. Preprocessing (outlier removal, missing value handling, derived metrics such as jitter and absolute values)
5. Feature extraction (CN0, CN0 variance, Noise CN0, DLL instability, Code jitter, Doppler jitter, correlator distortion)
6. Sliding window segmentation (1-second window, non-overlapping)
7. Feature aggregation (mean and variance per window)
8. Final structured dataset ready for machine learning

Use a clean academic style, labelled blocks, arrows between steps, minimal icons, white background, publication quality.

## **Appendix 2.** AI Prompt for Machine Learning Pipeline Figure

Create a high-resolution academic infographic (≥400 DPI, 1:1 aspect ratio) illustrating a machine learning model development and evaluation workflow for GNSS meaconing detection. Include the following steps:

1. Windowed feature dataset input
2. Labelling (0 = clean, 1 = meaconing)
3. Feature scaling
4. Model training (Random Forest, Support Vector Machine)
5. Blocked chronological validation (time-based split preserving temporal order)
6. Validation phase (threshold selection)
7. Testing phase (evaluation on unseen data)
8. Performance metrics (accuracy, precision, recall, F1-score, FAR, ROC-AUC, time-to-detection)
9. Additional analyses (learning curves, feature importance, ablation study, LOPO validation, time-split validation)

The diagram should be clean, structured, with clearly labelled blocks, arrows, and suitable for thesis publication.