Android GNSS Measurements under Spoofing and Interference

Andrea Botticella[*][†]
andrea.botticella@studenti.polito.it

Elia Innocenti[*][†]
elia.innocenti@studenti.polito.it

Renato Mignone[*][†]
renato.mignone@studenti.polito.it

Simone Romano[*][†]
simone.romano2@studenti.polito.it

ABSTRACT

This study explores the behavior of GNSS navigation on a commercial Android smartphone by analyzing raw measurements collected in both static and dynamic conditions. Using Google's GNSS Logger and MATLAB-based post-processing, we evaluate positioning accuracy, signal quality, and clock stability under normal operation and under spoofed inputs—where false positions and delays are injected into the data. The results highlight the vulnerability of smartphone GNSS to software spoofing and underline diagnostic features that may aid in anomaly detection and robustness improvement.

1 INTRODUCTION

Global Navigation Satellite Systems (GNSS) are fundamental to modern positioning services, widely embedded in smartphones and critical infrastructure. Despite their ubiquity, GNSS signals are inherently vulnerable to spoofing—an attack technique where counterfeit signals deceive the receiver into computing false location or timing information. To develop robust mitigation strategies, it is essential to understand how smartphone GNSS systems behave under both normal and manipulated conditions.

This work investigates the GNSS measurement behavior of a consumer-grade Android smartphone across different operating scenarios. We analyze baseline performance by collecting and processing raw GNSS data in two real-world environments: a static session on the rooftop of Monte dei Cappuccini and a dynamic session aboard a tram in urban Turin. In addition to this baseline study, we simulate spoofing by injecting false position inputs into the processing pipeline and, separately, by introducing artificial delays.

The remainder of this report is organized as follows.

- **Section 2** describes the experimental setup, including device configuration, data collection procedures, and the processing pipeline.
- **Section 3** presents results and discussion, contrasting static versus dynamic performance, examining spoofed-location impacts, and analyzing delay effects.
- **Section 4** summarizes the key findings and outlines directions for future work.

2 METHODS

2.1 Devices and Software

We used a Samsung Galaxy A51 with Android 11 for this experiment. GNSS Logger v3.1.0.4 was chosen due to its unrestricted access to raw GNSS measurements, compatibility with newer Android APIs, and ability to record detailed GNSS data that is suitable for precise analysis. MATLAB R2024b was employed to handle data because it comes with Google's GNSS toolbox, which accommodates robust analysis and visualization of GNSS measurements and position solutions.

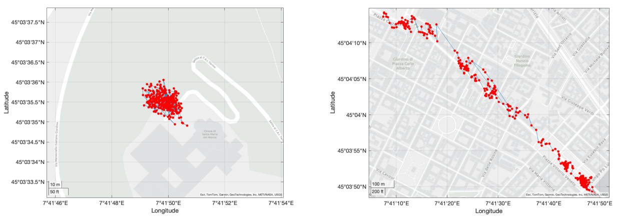
2.2 Data Collection Procedure

Two distinct 5-minute GNSS data logging sessions were conducted on 3 May 2025, under cloudy weather conditions, using the GNSS Logger app configured with the following settings enabled:

- **GNSS Location:** Enabled to capture location data.
- **GNSS Measurements:** Enabled to log raw GNSS measurements.
- **Navigation Messages:** Enabled to capture navigation data.
- **GnssStatus:** Enabled to log GNSS status information.
- **Sensors:** Enabled to capture sensor data.

The sessions were designed to capture both static and dynamic GNSS performance, with the following details:

- **Static Scenario:** Performed on the rooftop of Monte dei Cappuccini, Turin, starting at 10:35:20. The device was stationary throughout the entire session, providing baseline measurements.
- **Dynamic Scenario:** Conducted on tram line 15 from Piazza Castello to Piazza Vittorio Veneto, starting at 10:00:21, simulating a typical urban mobility scenario.



(a) Monte dei Cappuccini. **(b)** Tram Line 15.

Figure 1: Comparison of GNSS data: (a) Static scenario at Monte dei Cappuccini,(b) dynamic scenario along Tram Line 15.

2.3 Processing Pipeline

The raw GNSS data from the GNSS Logger served as the input dataset for MATLAB. Processing involved a scripted workflow via `ProcessGnssMeasScript.m`, where the following steps were executed:

1. **Filtering:** Data points with a carrier-to-noise ratio below 25 dB-Hz or satellite elevations below 15° were excluded to improve accuracy.

[*]The authors collaborated closely in developing this project.
[†]All the authors are students at Politecnico di Torino, Turin, Italy.

2. **Measurement Extraction:** Pseudorange and Doppler measurements were computed from GNSS timestamps and satellite transmission data.
3. **Weighted Least Squares (WLS) Positioning:** Applied to derive precise positioning and clock bias estimates.
4. **Visualization and Comparison:** Output plots from MATLAB, including pseudorange, pseudorange rates, and position solutions, were generated to facilitate comparative analysis of the static and dynamic scenarios.

Results from this processing pipeline provided insights into the differences in GNSS performance under static and dynamic conditions.

2.4 Spoofed-Input Configuration

Spoofing scenarios were emulated by introducing artificial variations to the recorded GNSS data through MATLAB processing. Specifically, mock positions were assigned by adjusting the parameter `spoof.position`, which represents modified latitude, longitude, and altitude coordinates. Additionally, artificial time delays were tested by adjusting the `spoof.delay` parameter, typically in milliseconds, to mimic delayed GNSS signal arrival. Such configurations facilitated evaluation of the impact of spoofing scenarios on position estimation reliability and accuracy.

2.5 Interference Scenario

Even a worst-case interference situation was simulated, replicating conditions in the vicinity of potential interference sources such as broadcasting antennas or communication areas of high density. GNSS data were collected near such sources of interference, and then processed with the same MATLAB procedure. Nominal condition comparison was established to analyze the impact of external interference on GNSS observables such as variation in pseudorange measurements, carrier-to-noise ratio, and positional accuracy overall.

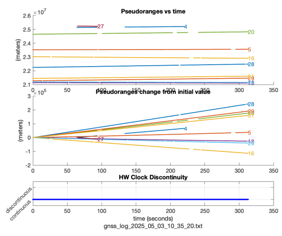
3 RESULTS AND DISCUSSIONS

3.1 Baseline Performance: Static vs. Dynamic

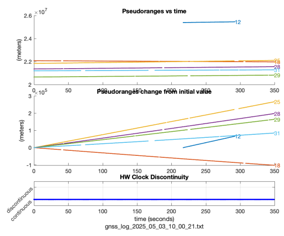
Comparing the static and dynamic scenarios highlights key differences in GNSS signal behavior, measurement stability, and receiver performance. Although both datasets were collected under similar atmospheric conditions, the receiver's motion in the dynamic case introduced visible changes across all GNSS indicators.

Pseudoranges vs Time. In the static case, pseudorange lines are mostly flat with occasional steps, reflecting steady satellite tracking and minimal receiver movement. In contrast, the dynamic case presents more pronounced variations, including jumps and slight trends, consistent with the receiver's motion on the Tram 15 route. SV 12 and SV 25 in particular show transient signal loss or handover events. These fluctuations reflect changes in satellite geometry and potential occlusions caused by urban structures.

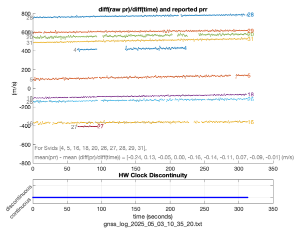
Pseudorange Change from Initial Value. The static dataset showed modest drifts in pseudoranges, largely attributable to receiver clock drift. The dynamic scenario, however, featured distinct upward and downward trends for different satellites (e.g., SV 25 increasing, SV 18 decreasing), corresponding to the receiver's motion relative to each satellite. This behavior confirms continuous positional change and introduces measurement dynamics absent in the static case.



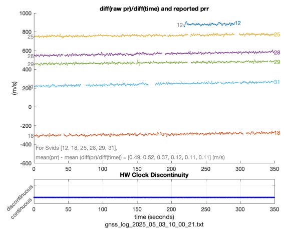
(a) Static: Pseudoranges vs time



(b) Dynamic: Pseudoranges vs time

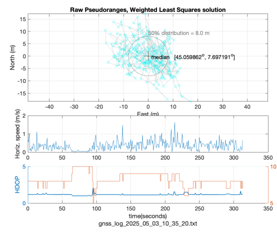


(a) Static: Δ Pseudorange

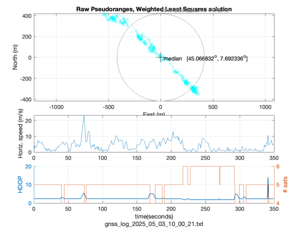


(b) Dynamic: Δ Pseudorange

Positioning and Speed. The position scatter in the static case remains tightly clustered, with near-zero horizontal speed, while the dynamic dataset exhibits a more dispersed cloud with higher velocity peaks (up to 20 m/s). These variations reflect real movement and acceleration events captured during Tram 15 trip from Piazza Castello to Piazza Vittorio Veneto. HDOP is comparably low in both scenarios, although it exhibits more fluctuation in the dynamic case, due to changing satellite visibility and geometry.



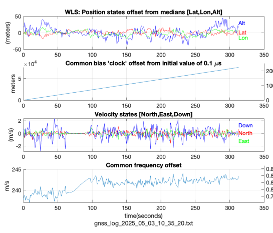
(a) Static: Position, Speed, HDOP



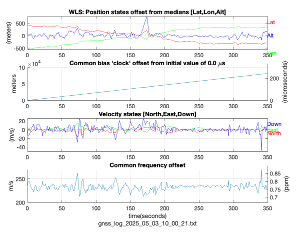
(b) Dynamic: Position, Speed, HDOP

State Offsets and Timing Bias. The static clock bias grows linearly, indicating thermal-induced drift. The dynamic scenario shows similar behavior, but with slightly more noise in the frequency offset and position offsets due to motion and rapid changes in the Doppler environment. Despite this, both datasets show eventual stabilization, suggesting effective internal correction mechanisms.

Overall, the dynamic dataset reveals the expected effects of motion on GNSS signals—greater variability, signal interruptions, and increased velocity states. In contrast, the static case offers a stable baseline, essential for identifying anomalies and for future spoofing detection tasks.



(a) Static: WLS states and bias



(b) Dynamic: WLS states and bias

3.2 Impact of Spoofed Position

In a spoofing attack, counterfeit signals are broadcast to mimic GNSS transmissions, often altering their timing, amplitude, or content. Typically, these deceptive signals are sent at higher power than the originals to mislead navigation systems. In our experiment, however, spoofing was implemented purely in software by adjusting the perceived reception time inside the professor's script to simulate spoofing behavior.

For the spoofed test we manually injected the coordinates of Piazza Vittorio Veneto (45.064749246294085, 7.6954660899754215) — a location adjacent to our true survey point — directly into the processing script. As shown in the "Spoofed Case" plots (see attached PDF), this causes the solver to converge exactly at the Piazza Vittorio Veneto coordinates rather than the true measurement site, introducing a horizontal displacement of several hundred metres.

Despite this spatial shift, all raw observables remain essentially unchanged relative to the normal run:

- **Carrier-to-noise density (C/N):** Histograms and time-series trace exactly overlap those of the baseline, confirming no change in received signal strength.
- **Dilution of precision (PDOP):** The satellite geometry quality curve is identical, indicating unchanged constellation geometry.
- **Pseudorange residuals:** The overall spread remains the same, but the entire residual distribution is offset by the constant delay corresponding to the spoofed displacement.

Additionally, the 50% (interquartile) range in the median-error distribution shrinks from approximately 8 m in the genuine case to about 4.7 m under spoofing. This reduction occurs because the solver consistently "locks" onto the spoofed coordinate, decreasing variability around that false point.

3.3 Effects of Timing Delays

In this delayed-spoofing scenario we again target the same true survey point but now introduce a software-only replay delay: the spoofer "listens" to genuine signals, waits 1 ms, then injects the spoofed Piazza Vittorio Veneto coordinates starting at 50 s into the run. The key parameters are set as:

```
cfg.delay   = 1e-3;
cfg.t_start = 50;

```

As visible in the plots, all observables remain nominal until t=50 s, at which point the solver's estimated position and receiver clock-bias exhibit a clear discontinuity as the spoof takes effect.

- **Position solution:** Prior to 50 s, the estimated coordinate coincides with the true static point. Immediately after 50 s, the solution jumps to the spoofed Piazza Vittorio Veneto location, replicating the static-spoof offset of several hundred metres.
- **Receiver clock-bias:** A sudden step of appears in the estimated receiver clock-bias track (Plot 11), directly corresponding to the 1 ms replay delay needed to shift the range solution by the planar offset.
- **Carrier-to-noise density (C/N):** The C/N time-series (Plot 9) shows no amplitude change at t=50 s—signal strength is unaffected by delay.
- **Dilution of precision (PDOP):** Satellite geometry quality (Plot 10) remains continuous and invariant through the spoof onset.
- **Pseudorange residuals:** Residuals (Plot 12) maintain the same spread, but their mean shifts abruptly at 50 s by the additional delay delta t.

From a solver-stability perspective, the sudden time bias causes an immediate step in both position and clock-bias, sometimes triggering a brief convergence glitch (extra iterations or loss of fix) as the estimator re-optimizes. Even a 1 ms replay delay can therefore introduce a large positional error and matching clock-bias shift without altering any standard observables. Detecting such covert delay-based spoofing thus requires monitoring for timing discontinuities.

3.4 Interference Effects

4 CONCLUSIONS

This work analyzed GNSS performance on a commercial smartphone under static, dynamic, and spoofed conditions using raw measurements and post-processing in MATLAB. In the static scenario, the device exhibited stable position estimates, low clock drift, and minimal variation in pseudorange and speed. The dynamic scenario, by contrast, showed increased noise in position, pseudorange rates, and clock bias—expected consequences of motion and satellite geometry changes.

The spoofing tests revealed that even simple, software-level injection of false coordinates can cause the GNSS solver to converge on a consistent but incorrect location, with little to no impact on first-order signal metrics like carrier-to-noise ratio or HDOP. This highlights a critical vulnerability: smartphone GNSS receivers can be misled without exhibiting obvious degradations in signal quality. Future extensions could include testing across multiple smartphones, introducing real-time spoofing detection mechanisms, and leveraging inertial or multi-constellation data for cross-validation. These steps would further explore the boundaries of GNSS resilience and help guide the development of more robust location-aware systems.

A APPENDIX

"Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat. Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum."