

Guardian Positioning System (GPS) for Location Based Services

Wenjie Liu

Networked Systems Security (NSS) Group
KTH Royal Institute of Technology
Stockholm, Sweden
wenjieli@kth.se

Panos Papadimitratos

Networked Systems Security (NSS) Group
KTH Royal Institute of Technology
Stockholm, Sweden
papadim@kth.se

ABSTRACT

Location-based service (LBS) applications proliferate and support transportation, entertainment, and more. Modern mobile platforms, with smartphones being a prominent example, rely on terrestrial and satellite infrastructures (e.g., global navigation satellite system (GNSS) and crowdsourced Wi-Fi, Bluetooth, cellular, and IP databases) for correct positioning. However, they are vulnerable to attacks that manipulate positions to control and undermine LBS functionality—thus enabling the scamming of users or services. Our work reveals that GNSS spoofing attacks succeed even though smartphones have multiple sources of positioning information. Moreover, that Wi-Fi spoofing attacks with GNSS jamming are surprisingly effective. More concerning is the evidence that sophisticated, coordinated spoofing attacks are highly effective. Attacks can target GNSS in combination with other positioning methods, thus defenses that assume that only GNSS is under attack cannot be effective. More so, resilient GNSS receivers and special-purpose antennas are not feasible on smartphones. To address this gap, we propose an extended receiver autonomous integrity monitoring (RAIM) framework that leverages the readily available, redundant, often so-called opportunistic positioning information on off-the-shelf platforms. We jointly use onboard sensors, terrestrial infrastructures, and GNSS. We show that our extended RAIM framework improves resilience against location spoofing, e.g., achieving a detection accuracy improvement of up to 24–58% compared to the state-of-the-art algorithms and location providers; detecting attacks within 5 seconds, with a low false positive rate.

CCS CONCEPTS

• Security and privacy → Software and application security.

KEYWORDS

Localization Attacks, Secure Localization, Geolocation APIs

ACM Reference Format:

Wenjie Liu and Panos Papadimitratos. 2025. Guardian Positioning System (GPS) for Location Based Services. In *Proceedings of the 18th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec 2025)*, June 30–July 3, 2025, Arlington, VA, USA. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3734477.3734707>

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

WiSec 2025, June 30–July 3, 2025, Arlington, VA, USA.

© 2025 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-1530-3/25/06.

<https://doi.org/10.1145/3734477.3734707>

1 INTRODUCTION

LBS are integral to daily life, relying on positioning provided by terrestrial and satellite infrastructures, e.g., cellular network (3/4/5G), Wi-Fi, Bluetooth, and GNSS (e.g., Global Positioning System (GPS)). Popular examples include navigation with Google Maps to a point of interest (POI), ride-hailing through Uber, and food delivery services. The correct position of the platform ensures the correct functionality of the application and the quality of the provided service.

Recent real-world vulnerabilities of LBS applications emerged [10, 39, 51, 52, 55], resulting even in scamming. Several attacks generate false position data fed into LBS: players of location-based games (Pokémon GO [39]); or scooter-sharing services and public transport, usually managed by geofencing or e-ticketing, virtually restricting the riding area or billing according to traveled distance [36, 55]. Position manipulation attacks can help win the game or break the geofencing, with the scooter seemingly within limits but in reality possibly far outside the fence, or allow traveling for free. Moreover, “ghost drivers” [52], automatically assigned to passengers based on spoofed positions, never physically reach passengers but only pretend picking them up, yet charge fees—perpetrating a taxi fee scam.

Position manipulation methods include GNSS spoofing, Wi-Fi spoofing, and virtual private network (VPN) proxies. GNSS spoofers [3, 40, 46, 54] broadcast adversarial satellite signals, either replayed or generated using open-source simulators [28], transmitted with higher power but in the correct format fool receivers to lock on to them instead of the actual GNSS ones. Wi-Fi geolocation, increasingly relevant in urban environments and indoor settings, often assisting GNSS, can also be manipulated. Attackers broadcast pre-recorded or downloaded Wi-Fi beacons, using consumer-grade Wi-Fi routers or low-cost Wi-Fi chips [18–20, 49, 50]. Cellular-based positioning based on base station signals, can also be attacked, by replayed signals or deploying rogue base stations [45]. IP geolocation methods (GeoIP) are also susceptible to manipulation, based on relays, transparent proxies, or VPNs to control round-trip time (RTT) and positioning [1, 26].

Highlighting the severity of LBS position manipulation, this study examines the impact of joint attacks targeting GNSS and other wireless signals and implements specific attacks. Specifically, we manipulate position estimates from different infrastructures in a coordinated manner. As a special case, given Wi-Fi beacons are a weak point, we jam GNSS and replay (or forge) Wi-Fi beacons to control positioning results.

Present solutions for GNSS spoofing detection, and more generally, secure positioning are mostly not designed for off-the-shelf platforms. They often rely on specialized hardware (e.g., resilient GNSS receivers or special-purpose antennas). For instance, a multi-antenna array is needed to calculate the angle of arrival (AoA) for

GNSS spoofing detection [6, 44]; or Wi-Fi channel state information (CSI) fingerprints require certain network interfaces to measure [29, 53]. Many modern smartphones have no such antennas or modules resilient to interference [7]. Even so, recent proposals thwart attacks based on the assumption that some other infrastructure is out of reach of the adversary. For example, [24, 30, 38] detect GNSS spoofing while assuming Wi-Fi or cellular signals are benign.

In response to these challenges, our solution leverages diverse opportunistic ranging and motion data sources to detect position manipulation attacks. While previous work often assumes certain signals are benign, our approach considers the possibility that all wireless signaling used for positioning may be compromised, leading to manipulation and disruption of LBS applications. By cross-validating opportunistic ranging information from GNSS and terrestrial network infrastructures, our approach is compatible and complements hardware fingerprinting or signal processing based attack detection [9, 29]. Additionally, it independently improves detection accuracy and mitigates the impact of LBS position manipulations from GNSS spoofing and rogue access points (APs).

We propose an extension of the RAIM technique, working on multiple subsets of distance estimates from satellites and network signals, then cross-validating position estimations on subsets of distances with onboard sensors. As it is almost impossible to jam all benign signals, there are almost surely benign subsets that can be the basis for detecting the attack. Our proposed method includes two main phases. First, it leverages ranging information—distances derived from GNSS signals and terrestrial infrastructures such as Wi-Fi and cellular. Subsets of ranging information are generated to compute multiple intermediate position estimates, each associated with uncertainty. To improve the efficiency and accuracy of this process, we incorporate a subset sampling strategy. The second phase is position fusion with onboard sensors to detect and mitigate attacks. By cross-validating intermediate position estimates, our algorithm identifies inconsistencies indicative of position manipulation, such as GNSS spoofing or rogue AP attacks. Unlike conventional detection methods that rely on a single infrastructure or unsecured fusion strategies, our scheme integrates diverse data sources in off-the-shelf platforms to improve robustness.

Our contributions are: We demonstrate geolocation API attacks and illustrate how they can disrupt LBS applications¹. Building on these insights, we develop a RAIM-based framework for detecting and mitigating threats in LBS position manipulation. Different from RAIM on GNSS, we use distance estimates from terrestrial networking infrastructures and GNSS with onboard sensors. By integrating this opportunistic ranging information from multiple sources, our approach gives an enhanced security likelihood against GNSS spoofing, rogue Wi-Fi APs, and other position manipulations. Our evaluation confirms the effectiveness of our approach, with improved true positive rate and delay in detecting attacks in various real-world scenarios.

In the rest of the paper: Section 2 provides background knowledge of LBS, GNSS spoofing, and Wi-Fi attacks. Section 3 presents our system model and adversary. Section 4 shows how to launch attacks on LBS with details in the appendix. Section 5 proposes our countermeasure. Section 6 evaluates the proposed scheme with

baseline methods, and Section 8 reviews related work about detection before we conclude in Section 9.

2 BACKGROUND AND PRELIMINARIES

2.1 Location-Based Services

LBS has reshaped industries ranging from navigation to marketing. Sensor fusion for precise and secure position estimation combines information from the inertial measurement unit (IMU), light detection and ranging (LiDAR), and network signals in vehicles or smartphones [30, 46]. Indoor navigation systems bridge the gap between outdoor and indoor environments, addressing GNSS limitations in indoor settings, using technologies such as Wi-Fi, Bluetooth beacons, and ultra-wideband (UWB) ranging [13]. Geofencing allows merchants to define virtual boundaries and trigger actions such as location-based messaging and dynamic pricing when users enter predefined areas [41]. In food delivery, LBS optimize shipping routes and reward participating taxis and couriers, enhancing efficiency [32]. Enabling targeted advertising, location-based marketing strategies have redefined advertising, with emphasis on contextual offers boosting engagement and conversion rates [27].

2.2 GNSS Spoofing Attacks

Spoofing GNSS involves transmitting fake but correctly formatted GNSS signals [21, 43, 47], to change position, timing, signal strength, and arrival angle, at the victim GNSS receiver, possibly hard to detect. Beyond efforts to authenticate GNSS signals and messages [11, 17] to mitigate spoofing attacks, adversaries can still record and replay authentic GNSS signals [28, 34]. Authentication is not yet widely supported by receivers and needs extra computational overhead and modification. A more advanced replay/relay attack [57] employs distance-decreasing attacks to tamper with signal timing, creating the false impression of earlier arrival. Recent research developed strategies to evade detection, such as slow variation to bypass tightly coupled GNSS/IMU systems [15, 46] or gradual spoofing algorithms targeting GNSS time [16].

2.3 Rogue Wi-Fi AP Spoofing

Rogue Wi-Fi APs are unauthorized devices posing significant cybersecurity threats [2]. These APs mimic legitimate APs and can intercept and alter client wireless communication, compromising integrity and confidentiality [48]. Wi-Fi attacks, including continuous or selective jamming and man-in-the-middle, have been demonstrated using commodity hardware in [50]. Manipulating received signal strength indicators (RSSIs) and related positioning statistics by rogue APs introduces position inaccuracies and inconsistencies [12, 19, 56]. Open-source tools for broadcasting Wi-Fi beacons can manipulate smartphone positioning results [20]. Furthermore, rogue APs deceiving Wi-Fi clients to automatically connect [33], can delay, relay, or replay client packets, thereby manipulating GeoIP positioning results [1, 26].

3 SYSTEM MODEL AND ADVERSARY

3.1 System Model

As shown in Figure 1, the system considers LBS deployed on mobile platforms (e.g., smartphone, tablet, or intelligent vehicle) that

¹<https://drive.google.com/drive/folders/1rZtwVYXi3OwKyS8Yzn23E2E18rBP-J3x>

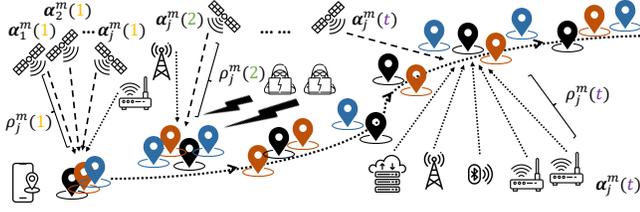


Figure 1: LBS applications have ranging information with anchor positions from GNSS and network infrastructures.

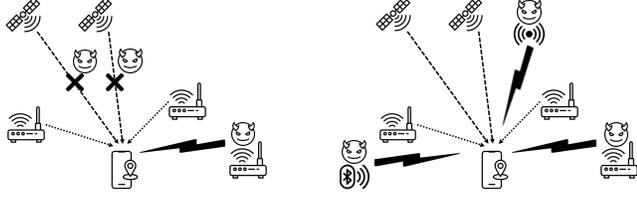


Figure 2: Left: Wi-Fi spoofing attack with GNSS jamming; right: coordinated location spoofing.

can process both GNSS and other opportunistic signals to position themselves (the devices). At time t , the true position of the platform, denoted as $\mathbf{p}_{\text{usr}}(t) \in \mathbb{R}^3$, is estimated by using GNSS and opportunistic information to compute $\mathbf{p}_{\text{lbs}}(t)$. Opportunistic information includes wireless signals received from network modules (e.g., Wi-Fi, cellular networks), Bluetooth, GeoIP, as well as motion data from onboard sensors (e.g., IMUs or wheel speed sensors).

Position estimates from different sources have varying accuracy; however, in benign conditions without any attacks, these estimates are expected to be consistent. If position manipulation occurs, significant deviations between $\mathbf{p}_{\text{lbs}}(t)$ and $\mathbf{p}_{\text{usr}}(t)$ are expected.

Notation: Denote motion measurements as velocity, $\mathbf{v}(t)$, acceleration, $\mathbf{a}(t)$, and orientation, $\boldsymbol{\omega}(t)$. Ranging information, denoted as $\rho_j^m(t)$, is associated with the positions of anchors $\boldsymbol{\alpha}_j^m(t)$; these are GNSS satellites, cellular base stations (BSs), Wi-Fi APs, Bluetooth devices, and GeoIP RTT servers; $j \in \mathcal{J}^m(t)$, where $\mathcal{J}^m(t)$ is the set of anchors at time t , $m = 1, 2, \dots, M$, and M is the number of opportunistic information sources.

3.2 Adversary Model

The adversary knows $\mathbf{p}_{\text{usr}}(t)$ and aims to control $\mathbf{p}_{\text{lbs}}(t)$, to disrupt the integrity and security of LBS. By manipulating wireless signals (e.g., GNSS pseudorandom noise codes, Wi-Fi beacons, cellular signals, Bluetooth beacons) and messages (e.g., RTT-related packets), it can compromise ranging information $\rho_j^m(t)$ used for positioning. We assume all wireless signals can be potentially attacked, although realistically, the adversary manipulates a subset of them at any given time. The following attack types are considered.

3.2.1 Wi-Fi Spoofing with GNSS Jamming. As GNSS spoofing attacks are feasible but still require a degree of sophistication, we propose jamming GNSS persistently, a relatively easy task, and force LBS to rely on network-based positioning information. The adversary injects falsified ranging information only by generating

and replaying Wi-Fi beacons captured from another place (as illustrated in the left part of Figure 2). These attack signals persist throughout the entire attack lifecycle but coexist with some legitimate transmissions, including original Wi-Fi and cellular signals.

3.2.2 Coordinated Location Spoofing. The adversary coordinates a sequence of falsified positions for the victim. It spoofs GNSS and replays cellular, Wi-Fi, and Bluetooth (and forges some of them, if possible), along the predetermined trace the attacker wishes to mislead the victim into perceiving. Through gradual deviation, the victim believes it is following a legitimate path. We assume that attack signals coexist with benign signals. Although the adversary may selectively jam Wi-Fi or cellular communication, it is infeasible to eliminate all legitimate transmissions at all times; it could also cause outages that would be easily detected.

4 ATTACK DEMONSTRATIONS

We demonstrate successful LBS position manipulation attacks based on the adversary model. All demonstrations were conducted with full attention to ethical considerations, in controlled environments with absolutely no impact on actual systems and users.

4.1 Multi-Band GNSS Spoofing

We demonstrate how a multi-band GNSS spoofing attack can manipulate fused location providers, such as Google Maps, with the attack applicable to other applications and services. The attack steps with detailed experimental configurations in Section 6.2.1 are:

Initial GNSS Jamming: Before spoofing, GNSS jamming forces the victim receiver to lose its lock on authentic signals. **Spoofing Signal Generation:** Using a GNSS signal generator (Skydel with USRP N310), the attacker broadcasts spoofing signals on multiple constellations and frequency bands (GPS L1/L5 and Galileo E1/E5a). These signals are transmitted at a higher power level than the real ones. **Dominant Position Estimation:** Although the smartphone location provider fuses data from GNSS, Wi-Fi, cellular, and Bluetooth, our attack ensures spoofing GNSS signals exhibit a favorable dilution of precision (DOP) to dominate the fusion algorithm.

Attack Results: Although the GNSS position is not consistent with the network-based positioning result, LBS applications directly provide the fused position near the spoofing position, thus imposing attacker control on the victim. If the user shares location data, the spoofed GNSS position will be used to train and build the network positioning and geolocation database of LBS providers. Note that attacking participatory sensing (e.g., on Google Maps [10]) can be done without physical presence and attack.

4.2 Wi-Fi Spoofing with GNSS Jamming

To overcome the high cost and portability issues of GNSS spoofers, we design a low-cost and easily deployable attack that leverages the replay of Wi-Fi beacons while in parallel jamming GNSS:

Enduring GNSS Jamming: Rather than a one-time jamming event, the adversary maintains GNSS jamming throughout the attack to prevent the reception of authentic GNSS signals. **Wi-Fi Beacon Replay:** A commercial Wi-Fi router (Linksys WRT1200AC) replays pre-recorded or artificially generated Wi-Fi beacons mapped to a predetermined spoofing position or trace. Unlike GNSS signals, Wi-Fi beacons do not require precise time synchronization, making

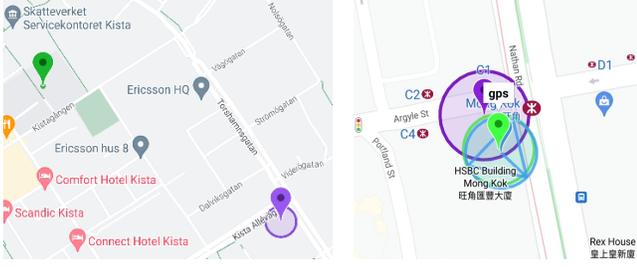


Figure 3: Left: Wi-Fi spoofing attack with GNSS jamming manipulates the application in the upper left corner (dark green pin) to the position at the bottom right (purple pin). Right: coordinated spoofing attack manipulates the network-based (purple circle with pin), GNSS (light green circle with pin), and fused (blue circle with pin) positions, deviating from Europe to Hong Kong.

them easier to manipulate. **Fallback Positioning:** With GNSS signals blocked, modern mobile devices rely on network-based positioning as a fallback. The replayed beacons thus mislead the fusion algorithm, causing it to compute a spoofed position.

Attack Results: We observe that both Android and iOS devices fail to notify users when positioning relies solely on (possibly adversarial) network signals, potentially exposing users to undetected LBS position manipulation. For instance, a taxi driver may falsify the driving route to get illegal profits or bypass trip security checking to take the passenger to an unintended destination without any notification. Figure 3 (left) and Appendix A illustrate the effectiveness of the position manipulation.

4.3 Coordinated Location Spoofing

Security-sensitive applications, such as mobile banking, typically validate location data by cross-referencing multiple sources (e.g., GNSS, Wi-Fi, and IP address). If the device meets verification criteria, region lock for the usage is lifted. Mobile payments, such as Revolut Card and WeChat Pay, have location-based security, comparing the mobile device position with the place of offline payment. If the two positions deviate, the transaction may be declined. To bypass them, as an example, given the attack is of broader interest, we designed a coordinated spoofing attack that manipulates all positioning.

Simultaneous Signal Manipulation: The attacker uses both a GNSS signal generator and a Wi-Fi router to broadcast spoofing signals to manipulate GNSS and network positions consistent with the spoofing position. **Network Beacon Crafting:** Wi-Fi beacons are replayed by using data extracted from public databases [8]. Since geolocation APIs do not verify the authenticity of beacons, the attacker can generate them without physical presence at the spoofing position. **Network Traffic Redirection:** The router relays all TCP and UDP packets from the connected devices to a cloud server located near the intended spoofing position using iptables.

Attack Results: By ensuring that the spoofing of GNSS, Wi-Fi, and (if applicable) Bluetooth signals all point to the same position, as shown in Figure 3 (right), our coordinated attack successfully bypasses cross-validation mechanisms deployed by secure applications. The detailed demonstrations are available in Appendix B.

Algorithm 1 Detection based on subset generation and cross-validation using available opportunistic information

Input $\{\alpha_j^m(t), \rho_j^m(t), \mathbf{v}(t), \mathbf{a}(t), \omega(t)\}, \mathbf{p}_{lbs}(t)$
Parameter Λ_f
Output *AttackDetected*

```

1: for  $t = 1, t++$  do                                ▶ Time index
2:   for  $m = 1, m++, m \leq M$  do                    ▶ Infrastructures
3:     for  $l = 1, l++, l \leq L^m(t)$  do                ▶ All subsets
4:        $\mathbf{p}_l^m(t) \leftarrow$  Section 5.1.3                ▶ Positioning
5:        $\hat{\mathbf{p}}_l^m(t) \leftarrow$  Section 5.2.1                ▶ Smoothing
6:        $\hat{\sigma}_l^m(t) \leftarrow$  Section 5.2.2                ▶ Uncertainties
7:        $f_{l,t}^m(\mathbf{p}) = \frac{1}{\hat{\sigma}_l^m(t)\sqrt{2\pi}} \exp\left(-\frac{1}{2}\left(\frac{\mathbf{p}-\hat{\mathbf{p}}_l^m(t)}{\hat{\sigma}_l^m(t)}\right)^2\right)$ 
                                                    ▶ Probability density function
8:     end for
9:   end for
10:   $f_t(\mathbf{p}) = 1 - \left(\prod_{m=1}^M \left(\prod_{l=1}^{L^m(t)} f_{l,t}^m(\mathbf{p})\right)^{\frac{1}{L^m(t)}}\right)^{\frac{1}{M}}$ 
                                                    ▶ Likelihood function and fusion
11:  if  $f_t(\mathbf{p}_{lbs}(t)) < \Lambda_f$  then
12:    AttackDetected = True
13:  else
14:    AttackDetected = False
15:  end if
16: end for

```

5 DEFENSE SCHEME

We propose a scheme easily deployable on mobile devices, using readily available opportunistic information, without assuming the presence of a trusted location source, considering that attackers can target multiple positioning methods simultaneously. Our scheme extends the RAIM method by leveraging redundant and opportunistic information to detect LBS position manipulation. It integrates multiple information sources, which include GNSS, signals of opportunity (SOP) (Wi-Fi, cellular, Bluetooth, GeoIP, etc.), and motion data (velocity, acceleration, and orientation based on on-board sensing), to strengthen detection. The overall process consists of two main steps: subset generation for positioning and position fusion for attack detection, as outlined in Algorithm 1.

First, real-time ranging information is collected from all available infrastructures. Through strategically combining different ranging information from heterogeneous infrastructures into subsets, it accommodates variations in signal characteristics, such as distance, uncertainty, and accuracy. As it is almost impossible to jam all benign anchors, it is reasonable to expect some benign subsets exist. For example, Wi-Fi jamming [50] cannot jam all beacons, as Wi-Fi channels are wide and relatively resistant to interference. To reduce computational complexity, as subset sizes range from the minimum required by the positioning algorithm to the maximum, our sampling strategy (Section 5.1.2) reduces the number of subsets and improves efficiency. Finally, intermediate position estimates are computed for each subset and infrastructure, using the corresponding positioning algorithms in Section 5.1.3.

Second, onboard sensors collect velocity, acceleration, and orientation data to refine intermediate position estimates obtained in the

first step. The proposed local polynomial regression in Section 5.2.1 smooths positions based on the physical movement constraints of the mobile platform. The filtered positions are then fused into probability density functions along with their associated uncertainties, as described in Section 5.2.2. A composite function in Section 5.2.3 is normalized to derive the final likelihood used for cross-validating position manipulations.

5.1 Subset Generation

5.1.1 Raw Data Preprocessing. The input data is the same crowd-sourced data used for LBS applications. Information from GNSS satellites, Wi-Fi APs, cellular BSs, Bluetooth devices, GeoIP RTT servers, and onboard sensors is recorded as they are available. GNSS signal data includes received times, satellite positions, and pseudoranges (estimates of satellite-receiver distances). Wi-Fi beacons include basic service set identifier (BSSID), service set identifier (SSID), and RSSI. Cellular data includes cell identifier, and RSSI. Bluetooth beacons include medium access control (MAC) and RSSI. GeoIP data includes the IP address of the platform and Internet Control Message Protocol (ICMP) messages containing RTT like ping values to data centers around the world. Onboard sensors provide motion measurements. All this information is timestamped and different types of data are temporally aligned.

GNSS pseudoranges, constant-biased approximation of the distance between the satellite and the receiver, are calculated by the time it takes for the signal to reach the receiver, multiplied by the speed of light. The pseudorange error may accumulate to hundreds of meters after a few seconds because the receiver clock quartz oscillator drifts. The clock is used to measure pseudoranges; thus, all pseudoranges have the same clock error factor. Then, in a benign environment, this ranging information can still accurately position the receiver by adopting at least four satellites, solving for its coordinates and clock error. Under a jamming attack, pseudoranges cannot be derived, partially or entirely. Under spoofing, pseudoranges of one or more constellations are modified by the attacker, i.e., the derived ranging information deviates from the benign one.

For Wi-Fi, cellular, and Bluetooth, the received opportunistic ranging information does not represent actual distances. Instead, it is RSSI, a negative number in dBm, which follows a log-distance path loss model. The closer the device, the stronger the signal, and vice versa. This ranging information is a relatively inaccurate approximation of distance compared to GNSS. Under jamming, the platform cannot receive valid packets, thus RSSI cannot be derived.

For GeoIP, both the IP address and ICMP based ranging are used. ICMP utilities (e.g., ping and traceroute) provide time delay (e.g., RTT). The delay multiplied by half the speed of light should be an approximation of the distance between the platform and the GeoIP RTT server. In the situation of delay manipulation attacks at the physical and data link layers (e.g., Wireshark), ICMP messages forwarded cause delay, thus longer ranges. In the situation of attacks using transparent proxy, this proxy is usually running on the network layer, so only TCP and UDP messages are managed rather than ICMP messages. Firewalls may be used to drop/reject ICMP messages, analogously to jamming, thus preventing ranging information from ping values.

Most importantly, we have a database of anchors, containing positions of Wi-Fi APs, cellular BSs, Bluetooth devices, and GeoIP servers, playing a role analogous to GPS ephemerides. Among all the anchors in the database, data cleaning can eliminate a majority of incorrect or non-fixed anchors, e.g., personal hotspots, Bluetooth headphones, and public transport Wi-Fi APs.

5.1.2 Subsets of Ranging Data. Subsets are generated to explore all possible combinations of anchors/constellations, from the minimum size required by positioning to the maximum. This process does not assume a specific number of attacked ranging information sources (e.g., spoofed constellations, rogue APs, and delayed RTTs), thus being applicable to any potential scenario.

The GNSS subsets include all possible combinations of constellations, including GPS, Galileo, GLONASS, and Beidou. For APs, BSs, or Bluetooth anchors, the receiver position can be determined using at least three RSSIs in trilateration, and the clock error cannot be estimated. Similarly, for GeoIP, at least three RTTs to determine the rough position. Hence, their number of subsets is $\sum_{i=3}^{J^m(t)} C(J^m(t), i)$, where $J^m(t) = |\mathcal{J}^m(t)|, \forall m > 1$. These subsets of ranging information (associated with anchor) indexes, j (from $\rho_j^m(t)$), are denoted as $S_l^m(t)$, where $l = 1, 2, \dots, L^m(t)$, with $L^m(t)$ the total number of subsets for the m th infrastructure (GNSS, Wi-Fi, etc.). Then, for each (l, m) , we use $\mathbf{p}_l^m(t)$ as the subset positioning result based on $S_l^m(t)$.

The number of generated subsets for localization may be very large, leading to sizable computational complexity; therefore, we use a subset sampling strategy: randomly select subsets before the next positioning step, with every subset selected or not via a predetermined probability distribution. For example, a discrete uniform distribution makes subsets equally likely to be chosen, not introducing bias or skew to $\hat{\mathbf{p}}_{\text{usr}}(t)$, thus does not undermine the cross-validation process. Through randomly choosing subsets, our detection scheme stays robust and adaptable to heterogeneous opportunistic information sources and attack types. Most significantly, it reduces the detection computational complexity.

5.1.3 Positioning Methods. In order to use the heterogeneous ranging information provided by multiple infrastructures, we have off-the-shelf positioning methods for the subsets from each data type, e.g., trilateration, multilateration, and geolocation localization. Each provides position estimation with an uncertainty value.

Trilateration for GNSS single point positioning is based on code observations of pseudoranges. The observations are affected by errors such as atmospheric delay, satellite clock, and receiver clock errors. GLONASS and GPS have differences in the way the ionospheric and tropospheric delays are modeled. Additionally, GLONASS uses a different frequency band than GPS, so the wavelength of the carrier wave is different. The pseudorange between the j th satellite and a user at $\mathbf{p}_{\text{usr}}(t)$ is $\rho_j^m(t) = \|\mathbf{p}_{\text{usr}}(t) - \boldsymbol{\alpha}_j^m(t)\| + \epsilon_n$, where ϵ_n models errors. Then, positioning uses the pseudoranges between the receiver and the satellite positions to compute the receiver position: $\|\hat{\mathbf{p}}_{\text{usr}}(t) - \boldsymbol{\alpha}_j^m(t)\| = \rho_j^m(t), j \in S_l^m(t)$.

Geolocation is distance-based positioning based on a weighted least squares problem to minimize the weighted sum of squared distances between anchors and the estimated device/user position [37]. These weights are determined based on the inverse square of

ranging: $\min_{\hat{\mathbf{p}}_{\text{usr}}(t)} \sum_{j \in \mathcal{S}_l^m(t)} \left(\|\hat{\mathbf{p}}_{\text{usr}}(t) - \boldsymbol{\alpha}_j^m(t)\| / \rho_j^m(t) \right)^2$, solved by numeric minimization algorithms such as the SciPy least squares.

GeoIP positioning combines both tabulation-based and delay-based IP geolocation [1]. Tabulation-based IP geolocation provides a lookup table to map IP address to an estimated position. Delay-based IP geolocation uses RTTs as ranging information with 10 to 20 anchors. It first maps RTT to distance based on a fitted function from training data. Then, the position is estimated as the centroid of the intersection of circles whose centers are the anchors and radii are the distances.

5.2 Position Fusion

5.2.1 Onboard Sensors. We use $\mathbf{p}_l^m(t)$ from Section 5.1 and $\mathbf{v}(t)$, $\mathbf{a}(t)$, $\boldsymbol{\omega}(t)$ from onboard sensors as input data, applying local polynomial regression to filter noise. Positions, $\mathbf{p}_{\text{usr}}(t)$, $\mathbf{p}_{\text{lbs}}(t)$, $\mathbf{p}_l^m(t)$, are represented in the format of the World Geodetic System 1984 (WGS84). $\boldsymbol{\omega}(t) \in \mathbb{R}^3$, from onboard sensors, comprises roll (ϕ), pitch (θ), and yaw (ψ) angles in the format of the sensor coordinate system of the mobile platform. The rotation matrix, \mathbf{R} , below converts a smartphone's sensor coordinates to WGS84:

$$\mathbf{R}(t) = \mathbf{R}_\phi(t) \mathbf{R}_\theta(t) \mathbf{R}_\psi(t) = \begin{bmatrix} \cos \psi(t) & -\sin \psi(t) & 0 \\ \sin \psi(t) & \cos \psi(t) & 0 \\ 0 & 0 & 1 \end{bmatrix} \\ \times \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta(t) & \sin \theta(t) \\ 0 & -\sin \theta(t) & \cos \theta(t) \end{bmatrix} \begin{bmatrix} \cos \phi(t) & 0 & -\sin \phi(t) \\ 0 & 1 & 0 \\ \sin \phi(t) & 0 & \cos \phi(t) \end{bmatrix}.$$

This definition differs from that used in aviation, with ϕ and θ interchanged, and ψ changes the direction of rotation. The state of the mobile platform, $(\mathbf{p}_{\text{usr}}(t), \mathbf{v}(t), \mathbf{a}(t))$, evolves over time, thus from $t-1$ to t , so

$$\mathbf{p}_{\text{usr}}(t) = \mathbf{p}_{\text{usr}}(t-1) + \mathbf{R}(t-1)\mathbf{v}(t-1) + \frac{1}{2}\mathbf{R}(t-1)\mathbf{a}(t-1) + \mathbf{n} \\ \mathbf{v}(t) = \mathbf{v}(t-1) + \mathbf{a}(t-1) + \mathbf{n}$$

where \mathbf{n} models noise. The state transition matrix is

$$\mathbf{F}(t) = \begin{bmatrix} \mathbf{1} & \mathbf{R}(t) \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \quad (1)$$

and the control-input matrix is $\mathbf{B}(t) = \begin{bmatrix} \frac{1}{2}\mathbf{R}(t) & \mathbf{1} \end{bmatrix}^\top$. We denote the estimated state after movement as

$$\begin{bmatrix} \hat{\mathbf{p}}_l^m(t) \\ \hat{\mathbf{v}}(t) \end{bmatrix} = \mathbf{F}(t-1) \cdot \begin{bmatrix} \mathbf{p}_l^m(t-1) \\ \mathbf{v}(t-1) \end{bmatrix} + \mathbf{B}(t-1) \cdot \mathbf{a}(t-1) \quad (2)$$

where $l = 1, 2, \dots, L^m(t)$, $m = 1, 2, \dots, M$. Then, to combine the motion with regression, we use an estimator $\hat{\mathbf{p}}_l^m(t) = \mathbf{W}\mathbf{t}$, where $\mathbf{W} \in \mathbb{R}^{3 \times (n+1)}$ is a matrix of polynomial coefficients, n is the order of the polynomial regression, \mathbf{t} is a $(n+1)$ dimensional vector, $[\mathbf{t}]_i = t^{i-1}$, and \mathbf{W} at (l, m, t) is calculated from the local polynomial regression problem:

$$\min_{\mathbf{W}} \sum_{t'=t-w}^t [\mathbf{W}\mathbf{t}' - \mathbf{p}_l^m(t')]^\top K_{\text{loc}}(t-t') [\mathbf{W}\mathbf{t}' - \mathbf{p}_l^m(t')] \quad (3) \\ \text{s.t.} \quad \|\mathbf{W}\mathbf{t} - \hat{\mathbf{p}}_l^m(t)\| \leq \epsilon_t$$

where w is a rolling window of the filter and $K_{\text{loc}}(t-t')$ is a kernel function assigning a scalar value to ensure the closer data has a higher weight, e.g., $\exp(-(t-t')^2)$. $\epsilon_t \in \mathbb{R}^3$ in the constraint is a

small tolerance for $\hat{\mathbf{p}}_l^m(t)$ to ensure the estimated position satisfies the movement in a short time duration. $\mathbf{p}_l^m(t')$ may not be available during the whole window w , then we use $\hat{\mathbf{p}}_{\text{usr}}(t')$ to complement missing positions.

The second derivative of the objective function in (3) with respect to \mathbf{W} is $2 \cdot \sum_{t'=t-w}^t K_{\text{loc}}(t-t') \cdot (\mathbf{t} \cdot \mathbf{t}^\top)^\top \otimes \mathbb{I}$, where \otimes is the Kronecker product. This derivative is always a positive definite matrix. In addition, the constraints are affine functions, so the problem is convex and solvable in polynomial time using Lagrange multipliers. After solving the problem, we have $\hat{\mathbf{p}}_l^m(t)$.

5.2.2 Uncertainty Modelling. The positioning uncertainty differs across various infrastructures and positioning methods. For GNSS-based trilateration, we use the position DOP metric. $(\sigma_x, \sigma_y, \sigma_z, \sigma_t) = \text{DOP} \triangleq \sqrt{\text{Tr}(\mathbf{Q})}$, where \mathbf{Q} is the covariance matrix of the least squares solution to the navigation equations and DOP is the square root of the trace of \mathbf{Q} . Then, the uncertainty $\hat{\boldsymbol{\sigma}}_l^m(t)$ is $(\sigma_x, \sigma_y, \sigma_z)$. For geolocation and other least squares algorithms, the uncertainty is represented by the residual of least squares. For other positioning techniques, we use the residual vector from the local polynomial regression to model the uncertainty of estimated positions.

5.2.3 Likelihood Function. The probability density function of each position estimate is defined as follows:

$$f_{l,t}^m(\mathbf{p}) = \frac{1}{\hat{\boldsymbol{\sigma}}_l^m(t) \sqrt{2\pi}} \exp\left(-\frac{1}{2} \left(\frac{\mathbf{p} - \hat{\mathbf{p}}_l^m(t)}{\hat{\boldsymbol{\sigma}}_l^m(t)} \right)^2\right) \quad (4)$$

where where $\hat{\mathbf{p}}_l^m(t)$ and $\hat{\boldsymbol{\sigma}}_l^m(t)$ denote the estimated mean position and standard deviation of the l -th estimate from the m th infrastructure, and all the operations are element-wise. Then, to aggregate intermediate position estimates along with their uncertainties, assumed to follow distributions $\mathcal{N}(\hat{\mathbf{p}}_l^m(t), \hat{\boldsymbol{\sigma}}_l^m(t)^2)$, we define a composite likelihood function. At time t , the likelihood function of $\mathbf{p}_{\text{lbs}}(t)$ under attack is computed as

$$f_t(\mathbf{p}_{\text{lbs}}(t)) = 1 - \left(\prod_{m=1}^M \left(\prod_{l=1}^{L^m(t)} f_{l,t}^m(\mathbf{p}_{\text{lbs}}(t)) \right)^{\frac{1}{L^m(t)}} \right)^{\frac{1}{M}} \quad (5)$$

where the cumulative term penalizes disagreement between $\mathbf{p}_{\text{lbs}}(t)$ and the fused distribution, yielding a higher likelihood when inconsistencies are detected. To determine whether position manipulation occurs, a threshold Λ_f is predefined based on established detection metrics. For example, the Z-score method sets the threshold by calculating the mean and standard deviation, while kernel density estimation non-parametrically estimates the probability density function to support threshold selection. We use receiver-operating characteristic curve (ROC) to analyze detection accuracy versus different false alarm rates in our evaluation results to assess the trade-off between them. If $f_t(\mathbf{p}_{\text{lbs}}(t))$ is larger than Λ_f , $\mathbf{p}_{\text{lbs}}(t)$ is deemed the result of an attack.

6 EVALUATION

We conducted the experiments in two settings and collected two distinct datasets to evaluate our detection approach. Jammertest 2024 [22] provides real-world attacks (fixed position, dynamic, time



Figure 4: A driving trace, GNSS positions, and network-based positions. The red dotted line frames the attacked area. The network signals are not specifically attacked.

jumping, etc.) solely on GNSS, but still allows for interesting results as the GNSS attacks affect other data. Coordinated location spoofing is done in a lab environment with more flexibility and control on the simulated attack strategies on GNSS and other positioning.

6.1 Dataset A: Jammertest 2024

To assess the real-world applicability of the proposed detection method against LBS position attacks, we collect data in an open-air environment. This setting provides a rare opportunity to legally observe a variety of GNSS attacks alongside benign wireless signals.

6.1.1 Experimental Setup. The test environment is an outdoor area in Bleik and its surroundings, with intermittent GNSS jamming and spoofing by the organizers of Jammertest [22]. The attack equipment involved various types of jammers, meaconers, and spoofers. Cigarette jammers, handheld jammers, and fixed jammers targeted GNSS as well as mobile communication bands (GSM and DCS). In the high power GNSS jamming scenario, jamming-to-signal ratio exceeded 24 dB over a distance of up to 73 km. Following successful GNSS jamming, GNSS spoofing was initiated. The fixed meaconer was deployed on a mountain, retransmitting live-sky signals from a long fiber-optic cable. The spoofing included stationary spoofing of small/large position jumps, simulated driving, flying spoofing, and more, employing Skydel with two USRP X300 software-defined radios (SDRs) to generate the GNSS signals following the pre-planned routes [22]. In most of the test cases, the location services on the smartphones were successfully deceived, as expected, with the position effectively manipulated. However, as shown in Figure 4, due to the intermittent nature of spoofing, signal blockage, environmental dynamics, and spoofing signals being mostly weaker than jamming, not all GNSS positions are spoofed.

6.1.2 Dataset Collection. We collected 68 driving traces using Android smartphones, including the Samsung S9, Redmi 9, Google Pixel 4 XL, and Pixel 8 (covering chipsets from Exynos, MTK, Qualcomm, and Google Tensor). Additionally, two u-blox receivers (ZED-F9P) served for ground truth positioning. The spoofing attacks did not target GLONASS, Beidou, and QZSS constellations, thus the ZED-F9P receivers were set up with clear views of these unaffected



Figure 5: The placement of the devices used for the coordinated location spoofing in NSS lab environment.

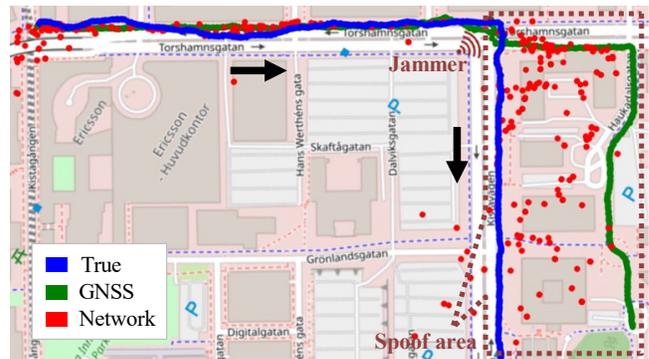


Figure 6: The actual walking trace, GNSS positions, and network-based positions. GNSS is spoofed, and network signals (cellular, Wi-Fi, and Bluetooth) are replayed in the red dotted box.

GNSS constellations, as well as an external GNSS reference station outside the affected region to ensure precise kinematic positioning.

Smartphones in the car captured potentially compromised GNSS and network signals. The GNSSLogger application recorded GNSS traces of the phone at a sampling rate of 1 Hz, consisting of RINEX and raw text files of satellite status with pseudoranges. The NetworkSurvey application recorded beacons and other messages from cellular, Wi-Fi, and Bluetooth anchors along the trace at approximately 0.3 Hz, in GeoPackage format, providing anchor names with RSSIs. In addition, acceleration (m/s^2), orientation (degrees), angular velocity ($^\circ/s$), and magnetic field (mT), are provided at 100 Hz from onboard sensors. Satellite positions were obtained from the broadcast ephemeris, and the positions of BSs, APs, and Bluetooth anchors were retrieved from a crowdsourced geolocation database calculated by the WiGLE.net application [8], based on both benign signals and measurements affected by adversaries or inadvertent contributors under GNSS attack.

6.2 Dataset B: Coordinated Attack

The coordinated location spoofing was conducted in our lab environment, mainly utilizing our in-house Skydel GNSS signal simulator and Ranatec RF shielded box. This setting manipulated wireless signals in the shielded box, while ensuring there is no unauthorized interference with actual users outside the lab facility.

6.2.1 Experimental Setup. We examined three walking traces, each approximately 1.5 kilometers in length, in Kista Science City and the streets of central Stockholm. The rationale of the test is to record GNSS and network signals of both benign and attack traces in advance and then superimpose GNSS spoofing, benign opportunistic ranging data, and adversarial ranging signals (collected via a dedicated trace at a distinct location/path, without any actual adversarial transmissions, thus no adverse effect). The simulator setup includes a workstation with Skydel 24.9.0, a Safran CDM-5 clock distribution module for synchronization, two Ettus USRP N310 SDRs, a Ranatec RI 187 RF shielded box, and a Tallysman TW7900P passive triple band GNSS antenna. Benign, spoofing, and jamming GNSS signals for GPS L1, Galileo E1, and BeiDou B1 are generated using Skydel, which streams IQ data to SDRs. N310 SDR outputs benign, spoofing, and jamming signals with gains of 60/65, 70, and 60 dB, respectively. A 40 dB attenuation is between the connection of the SDR output and the antenna input. Then, the antenna emits the generated GNSS signals in the shielded box. The placement of the devices is shown in Figure 5. The jammer enables CW, Chirp, Pulse, BPSK, BOC, and AWGN jamming at the central frequency of 1575.42 MHz with -15 dBm reference power. The spoofer uses -37 dBm reference power.

As shown in Figure 6, a jammer is simulated as placed in the middle of the victim’s walking trace, where the actual and the spoofed traces intersect. A spoofer is simulated to closely follow the victim on the actual walking trace. The attack strategy uses the static GNSS jammer to force the victim receiver to lose lock, and then, jamming is stopped and spoofing is launched for multiple constellations (GPS L1, Galileo E1, and BeiDou B1). Simultaneously, network-based positioning is targeted in a coordinated manner through the simultaneous, at this point, replay of cellular, Wi-Fi, and Bluetooth signals. These spoofed signals are introduced at the place where the true and spoofed positions initially coincide and then as they gradually deviate. Throughout the attack, benign network signals remain present. The actual trace of the victim in Figure 6 is the blue one, and the attacker imposes the misperception that the victim follows the green path by replaying actually recorded signals while walking across the green path. This can remain unperceived by a user for a significant amount of time.

6.2.2 Dataset Collection. We used a Redmi 9 Android smartphone to collect data. In the pre-collection phase, the GNSSLogger application was used to record GNSS traces along with onboard sensor measurements, and the NetworkSurvey application recorded messages from cellular, Wi-Fi, and Bluetooth. The sampling rate and format were consistent with Dataset A. In the post-collection phase, attacks were simulated using the smartphone and Skydel within the lab environment. Skydel replays the benign GNSS signals and generates spoofing signals according to the pre-collected traces. The GNSSLogger application records GNSS again in the shielded box. For network signals, the pre-collected network messages corresponded to the expected spoofing trace and were timewise aligned to the benign trace. We incorporate network messages of the spoofing trace into the benign trace emulating the attacker replaying network signals. The acquisition of satellite positions and the positions of BSs, APs, and Bluetooth anchors was performed in the same way as Dataset A.

6.3 Performance of Attack

During Jammertest 2024, not only were GNSS receivers affected, but also critical infrastructure, such as cellular BS timing, and crowd-sourced geolocation databases experienced failures. Network-based positioning dependent on crowdsourcing exhibited significant deviations: Most benign errors result in positions within 200 meters of the ground truth, whereas attack-induced errors predominantly range from 600 to 700 meters.

As we drove away from the jammer and spoofer, being protected by the terrain/buildings, we observed the affected receivers reacquired positions. Once we moved back to the jammer or spoofer line of sight, the receivers were once again affected. Furthermore, we observed that even when the attack targeted only a single constellation (e.g., GPS), its signals could impact antenna gain and then disable other constellations.

During our coordinated location spoofing experiments, with walk-based measurements and in-lab GNSS simulation, the simulated GNSS jammer and spoofer were static, with their path loss modeled. When the smartphone moved closer to the jammer, the GNSSLogger application indicated a decreasing carrier-to-noise power ratio. When the smartphone was around 10 meters from the jammer, its GNSS receiver completely lost lock on the benign signals. Then, we turned off the jammer and the spoofer began transmitting the generated GNSS signals corresponding to a falsified path. Due to the higher power level of the spoofing signals compared to benign ones, the smartphone GNSS acquired and locked onto the spoofing signals. The network signal spoofer was emulated at the data level by modifying recorded network traces. Cellular, Wi-Fi, and Bluetooth data from the benign and spoofing traces were merged together in the dataset, causing the network positions from the geolocation algorithm to deviate toward the intended spoofing trace, as per Figure 6.

Although the strategy using a GNSS jammer and GNSS spoofer demonstrated a high success rate in our experiments, we found it challenging to seamlessly spoof the smartphone without prior jamming but rather gradually increasing the spoofer signal power and drifting the receiver signal tracking.

6.4 Performance of Detection

The attack detection accuracy is evaluated by the true positive rate (P_{tp}) and the false positive rate (P_{fp}). P_{tp} represents the number of time intervals in which $p_{lbs}(t)$ is under attack and correctly detected to be so, over the total number of attack time intervals. Conversely, P_{fp} is the percentage of time intervals misclassified as being under attack while this is not so. Additionally, we define ΔT_d as the average attack detection latency, capturing the time elapsed from the onset of an attack to its detection.

Our baseline detection methods include one based on the Google Play Services fused location, the secure fusion-based GNSS attack detection in [30], one based on a Kalman filter, and one based on network-provided position. Google Play location is the most widely used fused location provider on Android devices. Secure fusion [30] assumes GNSS is possibly under attack and fuses GNSS, network, and onboard sensors. If the distance between this fused position and the raw GPS-provided position exceeds a threshold, the method flags the raw position as a spoofing. The Kalman filter

identifies position anomalies by analyzing the residuals, defined as the difference between the smoothed position and the raw GPS position. Network-based detection is derived from the geolocation algorithm leveraging network-based position only, also using distance discrepancy-based detection.

The experimental parameters were set as follows: $w = 20$, with a regression kernel $K_{\text{loc}}(t - t') = \exp\left(-0.3\left(\frac{t-t'}{w}\right)^2\right)$, a second-order polynomial regression model, and a sampling rate set to 50%. Spoofing labels are determined based on deviations: A positive label is assigned if the distance between the GNSS position and the ground truth is larger than 30 meters.

Figure 7 presents P_{tp} and P_{fp} results for Datasets A and B. Our proposed method improves from 18% to 24% for Dataset A and up to 58% for Dataset B in terms of P_{tp} , compared to Google Play location and network-based detection, when P_{fp} is between 5% and 10%. This confirms that as long as GNSS DOP is such that it leads to positions significantly more accurate than those based on network position, the Google Play location prioritizes GNSS position even when it is spoofed. It is important to note that Dataset B was collected in the shielded box, so it did not provide valid Google Play locations. Furthermore, network-based positioning lacks precision due to the corruption of geolocation databases caused by spoofing attacks in Dataset A. Additionally, coordinated location spoofing in Dataset B gradually increases the spoofing induced deviation, which limits the network position and Kalman filter based approaches to detect anomalies. Notably, P_{tp} is evaluated at the level of individual position fixes rather than over entire traces. P_{tp} represents the proportion of correctly identified spoofed intervals among all spoofed intervals, as opposed to ΔT_d measuring detection latency per trace.

Figure 8 shows ΔT_d versus P_{fp} . Our proposed method has up to 5 seconds gain for Dataset A in ΔT_d , when P_{fp} is between 5% and 10%. For our proposed detection, ΔT_d can detect spoofing attacks within an average of 4 seconds.

Compared with existing position fusion schemes and products, the proposed scheme can securely fuse position information, detecting and excluding malicious data. Unlike traditional fusion for secure localization [30], our scheme adopts a more fine-grained approach by analyzing individual ranging information for enhanced position security. However, the generation of subsets and the computation of positions for cross-validation are time-consuming. Hence, further effort on subset sampling strategies is needed.

6.5 Scheme Tuning

We analyze the sensitivity of our scheme to parameters, including sampling rate and window size. For the subset sampling strategy (Section 5.1.2), we evaluate the effect of different sampling ratios on P_{tp} . The results for Dataset A show that P_{tp} increases with higher sampling rates: With an increase from 0.25 to 1.0, P_{tp} improves from 86% to 90%, at $P_{\text{fp}} = 10\%$. This suggests that higher sampling rates capture a wider range of subsets, while at lower sampling rates the algorithm may miss important subsets. However, the accuracy is relatively insensitive to the sampling rate. When the sampling rate decreases from 1 to 0.25 (4 times less computation), P_{tp} is reduced by at most 4%. Although higher sampling rates generally lead to improved detection accuracy, practical constraints due to client computational resources and acceptable ΔT_d are important.

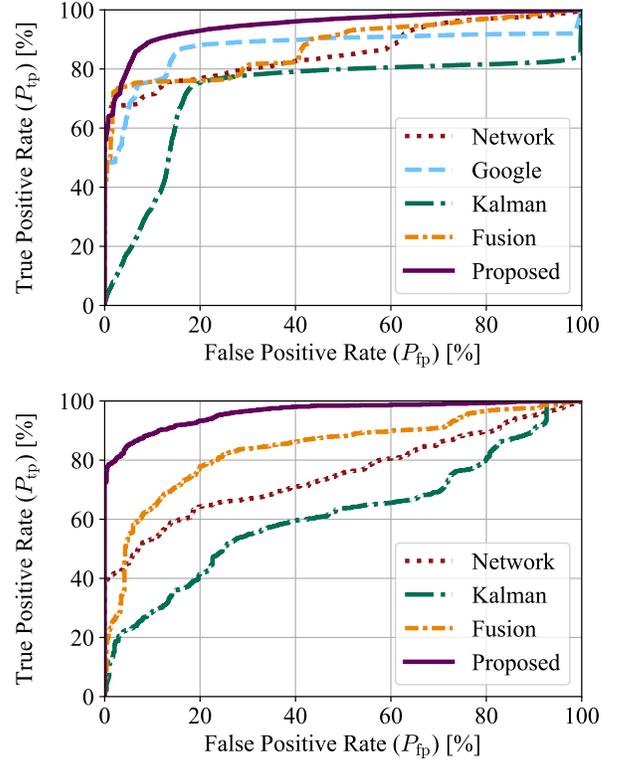


Figure 7: P_{tp} of the proposed and baseline methods for Dataset A (upper) and B (lower).

By adjusting w in (3), we evaluate the detection performance for different window sizes. Selecting an appropriate size requires a trade-off, as smaller w requires less computing power but potentially degrades detection accuracy. Conversely, oversized w may result in processing unnecessary historical data and increased computational complexity. Our results indicate relatively stable P_{tp} trending for $5 < w < 25$, but the computational time for detection when $w = 25$ is 1.7 times that of $w = 5$. In addition, P_{tp} for $15 < w < 25$ exhibits slightly improved performance compared to other window sizes; consequently, we selected this as the preferred range in our detector. We further conduct a small scale comparison among different orders of local polynomial regression in \mathbf{W} , showing that P_{tp} is higher for $n = 2$ compared to $n = 1$.

7 DISCUSSION

Ethical Concerns and Limitations: As GNSS spoofing is illegal, the experiments that led to Dataset A were conducted at Jammertest by the Norwegian authorities. For Dataset B, all spoofing signals were entirely contained within our RF shielded box, ensuring that they had no effect outside the lab. The adversarial trace collection for opportunistic signals was essentially a benign collection along an actual yet labelled and used, or the sake of emulating the attack, adversarial (to be imposed on the victim) path, with a superposition of this data with the benign one.

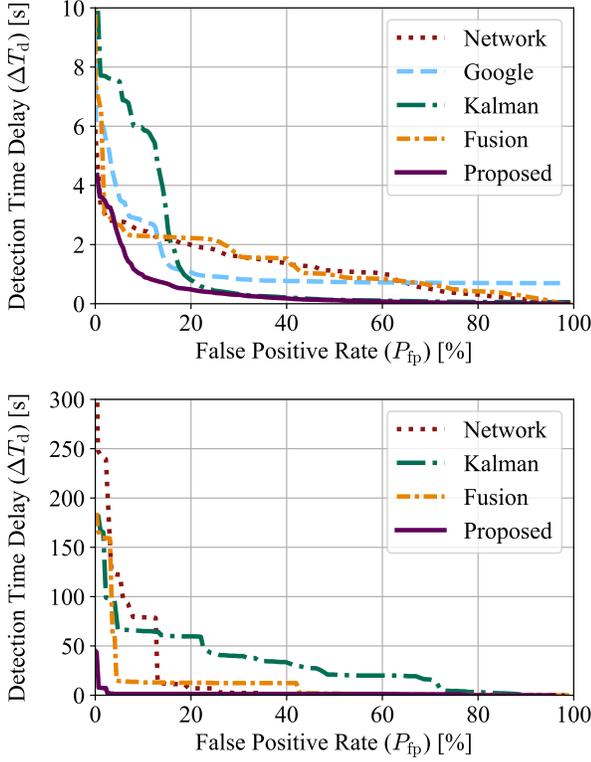


Figure 8: ΔT_d of the proposed and baseline methods for Dataset A (upper) and B (lower).

Alternative Data Cleaning: We explored large language model (LLM)-based data cleaning, since the Wi-Fi AP position data is crowdsourced from WiGLE.net [8] without quality assurance. The process of matching Wi-Fi AP SSID to their corresponding place names involves leveraging an LLM to extract text semantic information and utilizing POI APIs for mapping and validating. The following steps outline this data cleaning: Step 1 filters fixed places. It inputs the SSID name of the AP using the prompt “Is this Wi-Fi SSID from a static or mobile hotspot: SSID_NAME? Please answer static or mobile only.” This prompt directs the LLM to distinguish between fixed and mobile AP based on semantics. By filtering out mobile APs, we focus on identifying fixed places. Step 2 extracts keywords. It inputs the SSID name of the AP using the prompt “Then, can you extract some keywords of the place name from SSID_NAME? Please answer keywords directly.” This step prompts the LLM to extract relevant keywords indicative of the place name associated with the AP. Step 3 queries POI API based on the previously extracted keywords are used to obtain POI coordinates.

Processing Requirements and Overhead: We tested different brands of smartphones with chipsets from Exynos, MTK, Qualcomm, and Google Tensor. They all support logging GNSS pseudoranges at 1 Hz and network survey data every 3–10 seconds, depending on connectivity. Hence, detection should process the GNSS data within 1 second and the network data within 3 seconds. We ran the Python version of the algorithm on the aforementioned

MTK and Google Tensor platforms. Although the computation for the subsets took the longest time, each detection can be completed within 1 second without parallel optimizations. Even if the computing power of the mobile platform were not sufficient, we could still change the sampling rate in Section 5.1.2.

Deployment and Future Work: Although our experimental results and simulations show practical performance using opportunistic signals and consumer-grade sensors, the attack and detection test scenarios should be extended to cover, e.g., subtler, stealthier attacks, GNSS cold-start settings, static versus dynamic victim receivers, and scenarios with limited opportunistic information. The attacker can subtly change position, time, or signal power, in spite of the complexity of such attacks. Hence, integrating our detection with other lower-layer (position-, time-, and signal-based) spoofing detectors could provide a multi-layer defense. An attack during a cold-start period could impact GNSS performance, but our position consistency-based detection can remain robust. Future work should consider various deployment settings, including different hardware, user mobility, topology, and attack strategies.

8 RELATED WORK

RAIM Protecting GNSS: There are two primary forms of RAIM: residual-based and solution-separation [23]. Residual-based RAIM uses statistical hypothesis checking at the residual errors to identify potentially inaccurate measurements [25, 42]. The residuals can come from the least squares or Kalman filters: extended Kalman filter (EKF) RAIM makes use of sliding window filters to identify and eliminate outliers using GPS and inertial sensors [25, 42]. Solution-separation RAIM recursively assumes faulty satellites, generates subsets of the remaining satellites to derive solutions, and then excludes faults [31, 58]. For example, [58] integrates RANSAC clustering to classify position solutions.

Rogue Wi-Fi AP Detection: Detection of malicious Wi-Fi hotspots has received increasing interest. Some popular industry solutions [4, 14, 35] ignore all unknown APs or use AP MAC and SSID whitelists to prevent unauthorized APs. However, attackers can forge these records rather effortlessly. For example, most of the consumer-grade Wi-Fi routers can set any SSID, and open-source routers (e.g., OpenWrt) can even modify their MAC addresses. Hence, there is a need for detection beyond these Wi-Fi beacons. [29] uses wireless fingerprinting technology independent of client devices, but the robustness of fingerprinting gets worse in dynamic environments (rain, high traffic, etc.). In addition, semantic-based CSI in Internet-of-Things (IoT) environments offers potential accuracy advantages but requires specialized hardware and large-scale frequency band scanning [5].

9 CONCLUSION

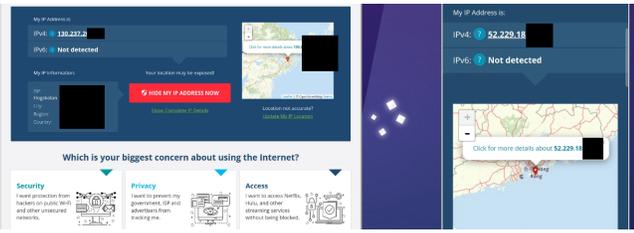
Our extended RAIM scheme detects LBS position manipulation using opportunistic ranging information and onboard sensor measurements: It cross-validates position estimates from different sources to give an attack likelihood. We implement position attacks in real-world LBS applications and show the feasibility of the proposed detection with experiments in various scenarios. The benefit is a significant reduction of both user and service exposure to scams without adding additional hardware.

ACKNOWLEDGMENTS

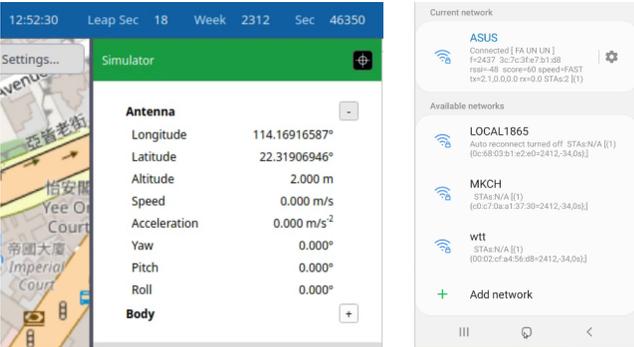
This work was supported in part by the SSF SURPRISE cybersecurity project, the Security Link strategic research center, and the China Scholarship Council. The computational resources were provided by NAISS, partially funded by the Swedish Research Council through grant agreement no. 2022-06725. We would also like to acknowledge the work of the organizers of Jammertest 2024.

REFERENCES

- [1] AbdelRahman Abdou, Ashraf Matrawy, and Paul C Van Oorschot. 2017. Accurate Manipulation Delay-Based Internet Geolocation. In *Proc. ACM Asia CCS*. Abu Dhabi, United Arab Emirates.
- [2] Bandar Alotaibi and Khaled Eleithy. 2016. Rogue Access Point Detection: Taxonomy, Challenges, Future Directions. *Wirel. Pers. Commun.* 90 (2016), 1261–1290.
- [3] Ala Altaweel, Hena Mulkath, and Ibrahim Kamel. 2023. GPS Spoofing Attacks FANETs: Systematic Literature Review. *IEEE Access* 11 (2023), 55233–55280.
- [4] ArubaOS. 2024. Detecting Rogue APs. *Hewlett Packard Enterprise Development LP* (2024). https://www.arubanetworks.com/techdocs/ArubaOS_64_Web_Help/Content/ArubaFrameStyles/New_WIP/Rogue_AP_Detection.htm
- [5] Ibrahim Ethem Bagci, Utz Roedig, Ivan Martinovic, Matthias Schulz, and Matthias Hollick. 2015. Using Channel State Information Tamper Detection Internet Things. In *Proc. 31st ACSAC*. Los Angeles, CA, USA.
- [6] Lu Bai, Chao Sun, Andrew G. Dempster, Hongbo Zhao, and Wenquan Feng. 2024. GNSS Spoofing Detection Mitigation Single 5G Base Station Aiding. *IEEE Trans. Aerosp. Electron. Syst.* 60, 4 (2024), 4601–4620.
- [7] Sean Barbeau. 2025. GPS Test Database. *Google Sheets* (2025). https://docs.google.com/spreadsheets/d/1jXtRC0EnnFNWj6_oFIVWflsf-b0jKfZpyhN-BXsv7uo/
- [8] Bobzilla, Arkasha, and Uhtu. 2023. Wigle.net. All Networks. Found Everyone. *Wigle* (2023). <https://wigle.net/>
- [9] Manuel Del Castillo. 2024. Protecting LBS Applications GNSS Spoofing. *Broadcom* (2024). <https://www.broadcom.com/blog/broadcom-bcm47765-gnss-antispoofing-receiver>
- [10] Cihan Eryonucu and Panos Papadimitratos. 2022. Sybil-Based Attacks Google Maps How Forge Image City Life. In *Proc. 15th ACM WiSec*. San Antonio, TX, USA.
- [11] Ignacio Fernández-Hernández, Vincent Rijmen, Gonzalo Seco-Granados, Javier Simon, Irma Rodríguez, and J David Calle. 2016. A Navigation Message Authentication Proposal Galileo Open Service. *J. Inst. Navigation* 63, 1 (2016), 85–102.
- [12] Manuel Flury, Marcin Poturalski, Panos Papadimitratos, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. 2010. Effectiveness Distance-Decreasing Attacks Impulse Radio Ranging. In *Proc. 3rd ACM WiSec*. Hoboken, NJ, USA.
- [13] Francesco Furfari, Antonino Crivello, Paolo Baronti, Paolo Barsocchi, Michele Girolami, Filippo Palumbo, Darwin Quezada-Gaibor, Germán M Mendoza Silva, and Joaquín Torres-Sospedra. 2021. Discovering Location Based Services: Unified Approach Heterogeneous Indoor Localization Systems. *Internet of Things* 13 (2021), 100334.
- [14] Douglas Gantenbein. 2024. Finding Remediating Rogue Access Points Microsoft Corporate Network. *Microsoft* (2024). <https://www.microsoft.com/insidetrack/blog/finding-rogue-access-points-on-the-microsoft-corporate-network/>
- [15] Yangjun Gao and Guangyun Li. 2022. A Slowly Varying Spoofing Algorithm Avoiding Tightly-Coupled GNSS/IMU Multiple Anti-Spoofing Techniques. *IEEE Trans. Veh. Technol.* 71, 8 (2022), 8864–8876.
- [16] Yangjun Gao and Guangyun Li. 2023. Two Time Spoofing Algorithms GNSS Receiver Instrumentation Modifying Satellite Clock Correction Parameters Navigation Message. *IEEE Trans. Instrum. Meas.* 72 (2023), 1–11.
- [17] GMV. 2023. Galileo Open Service Navigation Message Authentication. *Navigedia* (2023). <https://gssc.esa.int/navigedia/index.php/Galileo-Open-Service-Navigation-Message-Authentication>
- [18] Xiao Han, Junjie Xiong, Wenbo Shen, Zhuo Lu, and Yao Liu. 2022. Location Heartbleeding: Rise Wi-Fi Spoofing Attack Geolocation API. In *Proc. ACM CCS*. New York, NY, USA.
- [19] Xiao Han, Junjie Xiong, Wenbo Shen, Mingkui Wei, Shangqing Zhao, Zhuo Lu, and Yao Liu. 2024. The Perils Wi-Fi Spoofing Attack Geolocation API Its Defense. *IEEE Trans. Dependable Secure Comput.* (2024), 1–17.
- [20] Adam Harvey. 2016. Skylift: Wi-Fi Geolocation Spoofing ESP8266. <https://github.com/adamhrv/skylift>.
- [21] Todd E Humphreys, Brent M Ledvina, Mark L Psiaki, Brady W O'Hanlon, Paul M Kintner, et al. 2008. Assessing Spoofing Threat: Development Portable GPS Civilian Spoofers. In *Proc. 21st ION GNSS+*. Savannah, GA, USA.
- [22] Jammertest. 2024. The World's Largest Open Jamming Spoofing Test. *Jammertest* (2024). <https://jammertest.no/about-2/>
- [23] Mathieu Joerges, Fang-Cheng Chan, and Boris Pervan. 2014. Solution Separation Versus Residual-Based RAIM. *J. Inst. Navigation* 61, 4 (2014), 273–291.
- [24] Zaher M. Kassas, Joe Khalife, Ali A. Abdallah, and Chiawei Lee. 2022. I Am Not Afraid GPS Jammer: Resilient Navigation Signals Opportunity GPS-Denied Environments. *IEEE Aerosp. Electron. Syst. Mag.* 37, 7 (2022), 4–19.
- [25] Samer Khanafseh, Naeem Roshan, Steven Langel, Fang-Cheng Chan, Mathieu Joerges, and Boris Pervan. 2014. GPS Spoofing Detection Using RAIM INS Coupling. In *Proc. IEEE/ION PLANS*. Monterey, CA, USA.
- [26] Katharina Kohls and Claudia Diaz. 2022. VerLoc: Verifiable Localization Decentralized Systems. In *Proc. 31st USENIX Security*. Boston, MA, USA.
- [27] Cu Xuan Le and Hu Wang. 2020. Integrative Perceived Values Influencing Consumers' Attitude Behavioral Responses Toward Mobile Location-Based Advertising: Empirical Study Vietnam. *Asia Pac. J. Mark. Logist.* 33, 1 (2020), 275–295.
- [28] Malte Lenhart, Marco Spanghero, and Panos Papadimitratos. 2022. Distributed Mobile Message Level Relaying/replaying GNSS Signals. In *Proc. ION ITM*. Long Beach, CA, USA.
- [29] Yuxiang Lin, Yi Gao, Bingji Li, and Wei Dong. 2020. Accurate Robust Rogue Access Point Detection Client-Agnostic Wireless Fingerprinting. In *Proc. IEEE PerCom*. Austin, TX, USA.
- [30] Wenjie Liu and Panos Papadimitratos. 2023. Probabilistic Detection GNSS Spoofing Using Opportunistic Information. In *Proc. IEEE/ION PLANS*. Monterey, CA, USA.
- [31] Wenjie Liu and Panos Papadimitratos. 2024. Extending RAIM Gaussian Mixture Opportunistic Information. In *Proc. ION ITM*. Long Beach, CA, USA.
- [32] Yan Liu, Bin Guo, Chao Chen, He Du, Zhiwen Yu, Daqing Zhang, and Huadong Ma. 2018. FooDNet: Toward Optimized Food Delivery Network Based Spatial Crowdsourcing. *IEEE Trans. Mob. Comput.* 18, 6 (2018), 1288–1301.
- [33] Constantinos Louca, Adamantini Peratikou, and Stavros Stavrou. 2023. A Novel Evil Twin MiTM Attack 802.11 V Protocol Exploitation. *Comput. Secur.* 130 (2023), 103261.
- [34] Daniel Maier, Kathrin Frankl, Ronny Blum, Bernd Eissfeller, and Thomas Pany. 2018. Preliminary Assessment Vulnerability NMA-based Galileo Signals Special Class Record & Replay Spoofing Attacks. In *Proc. IEEE/ION PLANS*. Monterey, CA, USA.
- [35] IBM SevOne Network Performance Management. 2024. Wifi Monitoring Modern Networks. *IBM* (2024). <https://www.ibm.com/products/sevone-network-performance-management/wifi-monitoring>
- [36] Marazzi Michele, Patrick Jattke, Jason Zibung, and Kaveh Razavi. 2024. PayRide: Secure Transport e-Ticketing Untrusted Smartphone Location. In *Proc. DIMVA*. Lausanne, Switzerland.
- [37] Mozilla. 2023. Ichnaea. <https://github.com/mozilla/ichnaea/blob/main/ichnaea/api/locate/mac.py>.
- [38] Gabriele Oliveri, Savio Sciancalepore, Omar Adel Ibrahim, and Roberto Di Pietro. 2022. GPS Spoofing Detection Crowd-Sourced Information Connected Vehicles. *Comput. Netw.* 216 (2022), 109230.
- [39] Jeni Paay, Jesper Kjeldskov, Daniele Internicola, and Mikkel Thomasen. 2018. Motivations Practices Cheating Pokémon GO. In *Proc. 20th MobileHCI*. Barcelona, Spain.
- [40] Mark L Psiaki, Todd E Humphreys, and Brian Stauffer. 2016. Attackers Can Spoof Navigation Signals without Our Knowledge. Here's How Fight Back GPS Lies. *IEEE Spectrum* 53, 8 (2016), 26–53.
- [41] Sandro Rodriguez Garzon and Bersant Deva. 2014. Geofencing 2.0: Taking Location-Based Notifications Next Level. In *Proc. ACM UbiComp*. Seattle, WA, USA.
- [42] Paul F Roysdon and Jay A Farrell. 2017. GPS-INS Outlier Detection & Elimination Using Sliding Window Filter. In *Proc. Am. Control Conf.* Seattle, WA, USA.
- [43] Harshad Sathaye, Martin Strohmeier, Vincent Lenders, and Aanjan Ranganathan. 2022. An Experimental Study GPS Spoofing Takeover Attacks UAVs. In *Proc. 31st USENIX Security*. Boston, MA, USA.
- [44] Desmond Schmidt, Kenneth Radke, Seyit Camtepe, Ernest Foo, and Michal Ren. 2016. A Survey Analysis GNSS Spoofing Threat Countermeasures. *ACM Comput. Surv.* 48, 4 (2016), 1–31.
- [45] Altaf Shaik, Ravishankar Bargaonkar, Shinjo Park, and Jean-Pierre Seifert. 2018. On Impact Rogue Base Stations 4G/LTE Self Organizing Networks. In *Proc. 11th ACM WiSec*. Stockholm, Sweden.
- [46] Junjie Shen, Jun Yeon Won, Zeyuan Chen, and Qi Alfred Chen. 2020. Drift Devil: Security Multi-Sensor Fusion Based Localization High-Level Autonomous Driving GPS Spoofing. In *Proc. 29th USENIX Security*. virtual event.
- [47] Marco Spanghero and Panos Papadimitratos. 2023. Detecting GNSS Misbehavior Leveraging Secure Heterogeneous Time Sources. In *Proc. IEEE/ION PLANS*. Monterey, CA, USA.
- [48] Manesh Thankappan, Helena Rifá-Pous, and Carles Garrigues. 2022. Multi-Channel Man-in-the-Middle Attacks Protected Wi-Fi Networks: State Art Review. *Expert Syst. Appl.* 210 (2022), 118401.
- [49] Nils Ole Tippenhauer, Kasper Bonne Rasmussen, Christina Pöpper, and Srdjan Čapkun. 2009. Attacks Public WLAN-based Positioning Systems. In *Proc. 7th MobiSys*. Kraków, Poland.
- [50] Mathy Vanhoef and Frank Piessens. 2014. Advanced Wi-Fi Attacks Using Commodity Hardware. In *Proc. 30th ACSAC*. New Orleans, LA, USA.
- [51] Gang Wang, Bolun Wang, Tianyi Wang, Ana Nika, Haitao Zheng, and Ben Y Zhao. 2018. Ghost Riders: Sybil Attacks Crowdsourced Mobile Mapping Services.



(a) GeoIP manipulation by a Wi-Fi router relaying all TCP and UDP messages. Left: actual IP; right: manipulated.



(b) GNSS signal generator setting. (c) Generated beacons.

Figure 10: Settings of a coordinated attack.

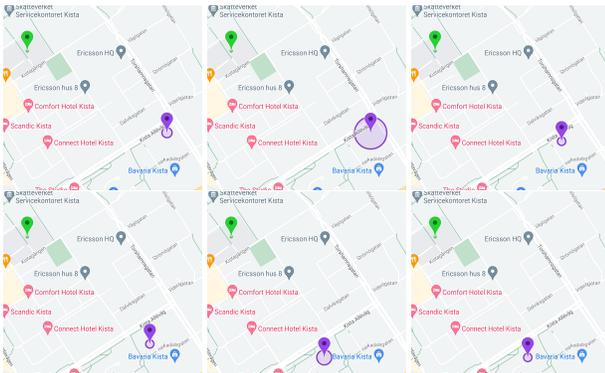


Figure 9: A time series of screenshots (from left to right, top to bottom) of a recorded attack video.

IEEE/ACM Trans. Netw. 26, 3 (2018), 1123–1136.

[52] Yue Wang. 2016. Ghost Drivers Are Just One Uber China’s Problems DIDI Takeover. *Forbes* (2016). <https://www.forbes.com/sites/ywang/2016/09/27/ghost-drivers-are-just...>

[53] Dawei Yan, Yubo Yan, Panlong Yang, Wen-Zhan Song, Xiang-Yang Li, and Pengfei Liu. 2022. Real-Time Identification Rogue WiFi Connections Wild. *IEEE Internet Things J.* 10, 7 (2022), 6042–6058.

[54] Jinghan Yang, Andrew Estornell, and Yevgeniy Vorobeychik. 2023. Location Spoofing Attacks Autonomous Fleets. In *Proc. VehicleSec*. San Diego, CA, USA.

[55] Ahmet Saim Yilmaz, Haydar Cukurtepe, and KUGU Emin. 2023. Geo-Location Spoofing E-Scoters; Threat Analysis Prevention Framework. *Balkan J. Electr. Comput. Eng.* 11, 4 (2023), 364–372.

[56] Lizhou Yuan, Yidan Hu, Yunzhi Li, Rui Zhang, Yanhao Zhang, and Terri Hedgpath. 2018. Secure RSS-fingerprint-based Indoor Positioning: Attacks Countermeasures. In *Proc. IEEE CNS*. Beijing, China.

[57] Kewei Zhang, Erik G Larsson, and Panos Papadimitratos. 2022. Protecting GNSS Open Service Navigation Message Authentication Distance-Decreasing Attacks. *IEEE Trans. Aerosp. Electron. Syst.* 58, 2 (2022), 1224–1240.

[58] Kewei Zhang and Panos Papadimitratos. 2019. Secure Multi-Constellation GNSS Receivers Clustering-Based Solution Separation Algorithm. In *Proc. IEEE Aerosp. Conf.* Big Sky, MT, USA.

A WI-FI SPOOFING WITH GNSS JAMMING

Figure 9 shows an application using the map service is fooled by a Wi-Fi router broadcasting beacons based on a list of SSID and BSSID along a pre-selected road. The purple pin is the estimated position from the application, but actually, the smartphone is located at the position of the dark green pin. We have also made two screen recordings demonstrating position manipulation under real GNSS jamming and Wi-Fi relaying in Bleik (available at: <https://drive.google.com/drive/folders/1rZtwVYX3OwKyS8Yzn23E2E18rBP-J3x>).

Listing 1: Wi-Fi location spoofing script tested on a consumer-grade router installed OpenWrt system and enabled virtual APs. It will broadcast Wi-Fi beacons of ap_set.

```
def uci_set_wifi (ap_set) :
    ssh.exec_command("sed -i '37,$d' /etc/config/wireless")
    for i in range(min(len(ap_set), 15)) :
        cmd_to_execute = "uci_batch <<_EOF\n"
        cmd_to_execute += f"""
            set wireless.wifinet{i}=wifi-iface
            set wireless.wifinet{i}.device='radio1'
            set wireless.wifinet{i}.mode='ap'
            set wireless.wifinet{i}.ssid='{ap_set.loc[i],
                "SSID"}'
            set wireless.wifinet{i}.encryption='psk2'
            set wireless.wifinet{i}.macaddr='{ap_set.loc[
                i, "BSSID"}'
            set wireless.wifinet{i}.key='Password'
            """
        cmd_to_execute += "commit\nEOF"
        ssh.exec_command(cmd_to_execute)
    ssh.exec_command("wifi_reload")
```

B COORDINATED LOCATION SPOOFING

This attack includes packets relaying for GeoIP manipulation, Wi-Fi location spoofing, and Skydel-based GNSS spoofing, as in Figure 10. The spoofed positions are coordinated, meaning the position of the relaying server and the coordinates in GNSS signal generator setting are near the pre-selected spoofed position. Moreover, the generated beacons mimic SSID and BSSID of Wi-Fi beacons.

Listing 2: The iptables rules for a proxy server running on the IP 127.0.0.1 and port 12345.

```
ip rule add fwmark 1 table 100
ip route add local 0.0.0.0/0 dev lo table 100
iptables -t mangle -N LAN
iptables -t mangle -A LAN -d 127.0.0.1/32 -j RETURN
iptables -t mangle -A LAN -d 224.0.0.0/4 -j RETURN
iptables -t mangle -A LAN -d 255.255.255.255/32 -j RETURN
iptables -t mangle -A LAN -d 192.168.0.0/16 -p tcp -j
RETURN
iptables -t mangle -A LAN -d 192.168.0.0/16 -p udp ! --
dport 53 -j RETURN
iptables -t mangle -A LAN -j RETURN -m mark --mark 0xff
iptables -t mangle -A LAN -p udp -j TPROXY --on-ip
127.0.0.1 --on-port 12345 --tproxy-mark 1
iptables -t mangle -A LAN -p tcp -j TPROXY --on-ip
127.0.0.1 --on-port 12345 --tproxy-mark 1
iptables -t mangle -A PREROUTING -j LAN
```