GhostBuster:
Detecting Misbehaving Remote ID-enabled Drones

Mart Keizer*, Savio Sciancalepore*, Gabriele Oligeri†
*Eindhoven University of Technology – Eindhoven, Netherlands
m.j.b.keizer@student.tue.nl, s.sciancalepore@tue.nl
†Division of Information and Computing Technology
College of Science and Engineering, Hamad Bin Khalifa University, Qatar Foundation, Doha, Qatar.
goligeri@hbku.edu.qa

***Abstract*—[1]Remote ID (RID) regulations soon applicable worldwide force drones to broadcast plaintext wireless messages providing, among others, their current location. However, malicious drone operators who want to stay stealthy might disclose RID messages carrying out location spoofing attacks, i.e., report forged locations, different from the actual ones. In this paper, we investigate the feasibility of using wireless localization approaches to detect drones carrying out location spoofing attacks. To this aim, we propose GhostBuster, a modular solution for detecting misbehaving RID-enabled drones, and we evaluate its performance via an extensive experimental campaign based on open-source data from actual drone flights. Through the analysis of real data in an area of $1.5km \times 2.5km$, we show that systems integrating multiple receivers can take advantage of multiple RID messages to verify the location reported by RID-enabled drones with a success rate of 95% up to 364 meters with 12 receivers. We also show that channel conditions play a crucial role in defining the maximum achievable spoofing detection performance.**

***Index Terms*—Drones Security; Mobile Security; Location Verification.**

I. Introduction

The enormous increase in the number of drones operated daily forced regulatory bodies around the world to design rules to regulate drone traffic [1]. In the United States (US), the Federal Aviation Administration (FAA) promulgated the Remote Identification of Unmanned Vehicles (RID) regulation [2]; in the European Union (EU), the European Aviation Safety Agency (EASA) published the amended regulations for remote identification 2020/1058 [3], and other similar initiatives apply also for China, Japan and India [1]. All such regulations require almost any drone (with a few exceptions, see Sect. III) to regularly broadcast wireless messages including their identity, current location, and location of the Ground Control Station (GCS), to name a few [2].

Although effective on paper, RID regulations cannot deal with the inherent heterogeneity and complexity of the drone ecosystem, especially compared to traditional aviation. In fact, aircraft are typically very expensive and operate according to strict rules. Instead, drones can be very cheap and are typically operated by potentially unqualified and untrusted users, who can use them for malicious purposes [4]. Malicious drone operators may easily stop transmitting RID messages. In such a scenario, regular drone detection systems have already shown remarkable performance in identifying malicious drones [5]. Alternatively, stealthy attackers using drones might continue to broadcast RID messages, but at the same time carry out *location spoofing attacks*, i.e., report a falsified drone location so as not to trigger alarms on location monitoring systems [6].

Wireless localization solutions might be applied to check the consistency of the location reported through RID messages with the one estimated locally through one or multiple sensors [7]. However, to the best of our knowledge, there are no studies specifically investigating the capabilities of wireless localization solutions and drone detection systems to identify drones that carry out location spoofing attacks via RID messages. Some solutions are available from research carried out for aviation security (see Sect. II). However, traditional recipes for aircraft security do not apply to drone security, due to the cited differences in the ownership of flying vehicles, the coverage of the adopted wireless communication technologies, the transmission power, the height of the flying vehicles, and the range of drone operations.

Contribution. In this paper, we investigate the feasibility of using wireless localization solutions to detect RID-enabled drones carrying out location spoofing attacks. To this end, we design *GhostBuster*, a new ad-hoc solution to detect drones that falsify the location reported in RID messages. Our solution relies on the comparison of the location reported by drones within RID messages to the location estimated via local sensors and raises an alarm when the difference between the two cited locations is anomalous. As a distinctive feature, GhostBuster can be adopted either by standalone users (equipped with a single receiver) or by safety-critical deployments, i.e., Critical Infrastructures (CIs), featuring a network of multiple sensors. We report the results of an extensive performance assessment of our solution, both through simulations and analysis of real drone flight data. We show that, with $K = 12$ receivers, GhostBuster can detect with 95 % success rate drones that spoof their location at a minimum distance of 364 m, and such performances improve further with better

[1] This is a personal copy of the authors. Not for redistribution. The final version of the paper will be available soon through the digital library IEEExplore.

data and channel conditions. Overall, our work contributes to improving the effectiveness of malicious drone detection and makes a step toward the safe and secure integration of unmanned aircraft into everyday life.

Roadmap. The rest of this paper is organized as follows. Sect. II reviews related work, Sect. IV introduces the reference scenario and adversary model, Sect. V introduces GhostBuster, Sect. VI reports our extensive performance assessment, and finally, Sect. VII reports conclusions and future work.

II. Related Work

Many solutions for wireless Radio Frequency (RF) localization are available for a variety of use cases and communication technologies [7]. In general, two main sources of information can be used, i.e., the Received Signal Strength (RSS) and the Time of Arrival (ToA) of wireless signals. ToA approaches use the timestamps of the RF messages at a set of receivers to estimate the location of the transmitter. They have been used, e.g., in wireless networks and underwater communications [8], [9]. However, such techniques require either a network of time-synchronized devices or the usage of dedicated synchronization techniques. In addition, they require very precise timestamps (at the nanosecond scale) and a minimum number of 4 devices to be able to localize a target, being generally more expensive. RSS-based techniques use the received power level of the messages at one or more receivers to estimate the location of the transmitter. Compared to ToA approaches, RSS-based ones do not require synchronizing the receivers, so being often easier to deploy [10], [11]. At the same time, RSS-based approaches are usually less precise than time-based ones, being much more subject to fast fading and noise (interferences).

In this context, only a few works investigated location verification of a non-cooperative remote wireless transmitter. In particular, the problem considered in this work for RID messages emitted by drones shares similarities with the verification of the location reported by aircrafts broadcasting Automatic Depedent Surveillance - Broadcast (ADS-B) messages. In that research area, many contributions are available based on the usage of ToA measurements [12], Doppler shift [13], predicted trajectory information [14], and time ranges [15]. However, note that the features of the drone ecosystem are very different from those of the aircraft. First, the transmission range of RID WiFi messages (up to 15 km) is much lower than the one of ADS-B messages (500 km), affecting the area where receivers can be deployed and making many physical-layer solutions (such as Doppler Shift) hardly applicable. Moreover, while aircraft trajectories are usually well known, drone flight is often unpredictable, preventing the use of publicly-available trajectory information. Also, the WiFi channel (2.4 GHz) is much noisier and subject to interference than the ADS-B channel (1.090 GHz), making these solutions potentially less efficient. On top of this, drones usually fly at much lower altitudes than aircraft. Thus, the signal is likely more exposed to shadowing and multipath. Therefore, the solutions and results achieved in the ADS-B domain do not directly map to the drone ecosystem. Such considerations motivate the investigation of the deception capabilities of RID-equipped drones, making our study novel and interesting on its own.

III. Remote Identification of Unmanned Vehicles

The RID rule has been introduced by the US-based FAA in 2021 and is expected to become effective for US airspace in September 2023 [2]. To increase the control of airspace, RID forces all Unmanned Aircrafts (UAs) to periodically broadcast information on the wireless channel regarding their unique identifier, current location, location of the GCS, current timestamp and emergency status [1]. Drones emit such information as plain text (no mandatory message encryption nor authentication) with a message rate of 1 per second, using WiFi or Bluetooth [16]. Whenever a drone is not equipped with compatible communication technology, it should be retrofitted with external modules to comply with the regulation. Moreover, according to clause 89.310(g) [17], the broadcasting device must optimize its broadcast range, i.e., use the maximum available transmission power. The described operating mode is known as *broadcast mode*. RID also allows for an optional *network mode*, where RID messages are delivered over the Internet to a specific public network address.

Drone remote id protocol (drip). The RID rule only provided the requirements that drones have to satisfy, but not the architecture and security objectives. Such aspects have been the focus of the drip Working Group (WG) by at the Internet Engineering Task Force (IETF). The goal of drip is to make RID realizable in practice and reliable, particularly in emergency situations [18]. The WG formulated several Request For Comment (RFC) documents, e.g., for drone requirements, security and privacy issues. To the aim of this manuscript, we focus on the reference architecture described in [19]. Accordingly, we denote the drone as the UA and the wireless receivers as *observers*, distinguishing between *Public Safety Observers*, deployed e.g. by a CI, and *General Public Observers*, i.e. standalone users interested in verifying drones' reported location. As a result, our solution is fully compliant with the architecture, notation, and requirements of drip.

IV. System and Adversary Model

A. System Model

Fig. 1 shows the reference scenario considered in this work. We consider K wireless RF receivers, namely *observers*. In line with the IETF WG *drip*, such observers can be either *General Public Observers*, i.e., standalone receivers deployed by users to monitor drone traffic for amateur purposes, or *Public Safety Observers*, deployed by a system owner, e.g., a CI operator, around a sensitive target to monitor the wireless RF spectrum in a given area of interest for public safety reasons. For the most general case of *Public Safety Observers*, we do not assume any synchronization in place between the observers, nor any mutual coordination. Each observer monitors the Industrial Scientific Medical (ISM) frequency band $[2.4 - 2.5]$ GHz and, when detecting any RID packet, it logs the reported location and the RSS. We consider the

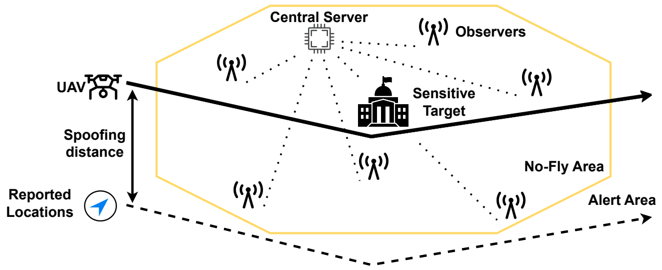

Fig. 1. Reference Scenario. The observer(s) use the RSS of received RID packets to verify that the reported locations match the estimated ones.

TABLE I
NOTATION AND BRIEF DESCRIPTION.

Notation	Description
K	Number of observers.
P_T	(Maximum Possible) Transmission power used by the drone.
N	Number of RID messages used for location verification.
$lat_{i,t}$	Latitude reported by the drone i at time t.
$lon_{i,t}$	Longitude reported by the drone i at time t.
$alt_{i,t}$	Altitude reported by the drone i at time t.
K	Number of RID messages used for localization.
τ	Threshold for detecting anomalous drones.
ϵ	Adversary.

observers to be able to extract the RSS of a received packet. We do not care how RSS values are computed (either on the whole packet or on the preamble of the packet). Finally, without loss of generality, we consider observers capable of logging the RSS as an integer value, with no decimal digits. Such an assumption allows us to consider the worst case of low-cost receivers. We discuss the usage of more expensive receivers in Sect. VI-D. In line with the regulations enforced by the vast majority of CIs, we consider the existence of a *no-fly area* around the location of the sensitive target (yellow line in Fig. 1), where UAs cannot fly. However, the combined detection range of all the observers is usually larger than the no-fly area, including also areas where drones are allowed to fly, possibly with some restrictions. We denote such area as the *alert area*. For the case of K public safety observers, we assume that each observer is connected (wired or wireless) to a *Central Server*, where it can report data.

Our scenario also assumes the presence of one or more UAs, i.e., drones. We consider the drones to be compliant with the RID rule described in Sect. III. Thus, once every second, they broadcast messages on the ISM band, reporting their unique identifier, location, location of the GCS, timestamp, and emergency status. According to the requirements of RID, we consider drones that emit messages with the maximum available transmission power, namely P_T. To ensure compliance, the UA is equipped with a Global Navigation Satellite System (GNSS) receiver, which is used to reliably estimate both the current location (in terms of latitude $lat_{i,t}$, longitude $lon_{i,t}$, and altitude $alt_{i,t}$) and local time t_s. We do not make assumptions about the existence and nature of remote control of the UA, which can be either remotely-piloted or (semi)-autonomous. For the reader's convenience, we report the main notation used throughout this manuscript in Tab. I, together with a concise description.

In this context, our investigation aims to let observers collaboratively detect UAs falsifying the current location as reported within RID messages. We provide more details on the adversary model below.

B. Adversary Model

In this manuscript, we assume an adversary ϵ, which deploys a drone to fly over the area monitored by the observers described in Sec. IV-A. The objective of the adversary is to fly over the no-fly area of the CI without being detected by the network of observers deployed by the CI operators. To this aim, ϵ can forge the location of the UA reported in the RID messages (namely, the *Reported Location* in Fig. 1), shifting it from the actual location by a given distance (namely, the *Spoofing Distance* in Fig. 1), to appear at another location, outside of the no-fly area. In Sect. VI-D, we also take into account the ability of the adversary to transmit messages at a power $P_{i,t}$ different from the maximum.

Note that a *naive* adversary might achieve the objective (undetected violation of no-fly area) by turning off the transmission of RID messages. Such an adversary model is not in the scope of our work and falls into the general topic of drone detection, which has been widely investigated in the literature. On the contrary, our adversary model allows the drone to still comply with current RID regulations. Such behavior can possibly be perceived as non-malicious while, in fact, abusing RID to perform malicious actions. In this context, our adversary is smarter and stealthier, at the same time, than one simply turning RID off.

At the same time, we consider the observers and the central server *trusted entities*. Thus, we do not consider attacks where ϵ compromises observers, e.g., carrying out replay attacks.

Finally, note that in the general case, ϵ does not know the location where the observers are placed. In Sect. VI-D, we discuss the implications of partial or full knowledge of the deployment location of the observers.

V. DETECTION OF MISBEHAVING REMOTEID-ENABLED DRONES

A. Our Solution at a Glance

Fig. 2 reports an overview of our proposed approach.

GhostBuster consists of two phases: training and deployment. Our solution relies on the deployment of K observers in the monitored area. Such observers detect and decode RID packets emitted by drones in the monitored area and compute a set of *ranges*, i.e., estimated distances (one for each packet) of the receiver from the transmitter location. We denote by N the total number of packets delivered by the drone in a given time frame T. Each observer delivers the estimated ranges to the central server, which combines the ranges and estimates the location of the transmitter. At training time, we acquire a set of

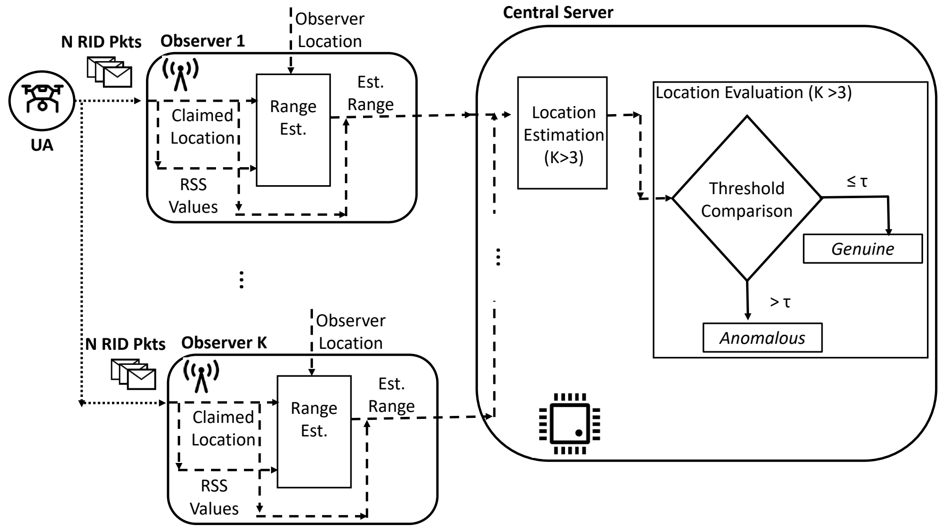

Fig. 2. GhostBuster at a glance. One or more observers (K) acquire N RID packets emitted by drones, estimate their distances from the drone, and submit such values to the central server for comparison with the reported locations by means of a threshold value τ.

differences between the actual location(s) of the transmitting drones and the location(s) estimated through our framework, which constitutes the expected profile of such differences. The output of such a training phase is a detection threshold τ, i.e. the maximum error (in meters) between the estimated location and the actual location. At deployment time, the estimated location(s) are compared to the location(s) claimed by the drone in the RID messages. If the distance between the two is greater than the threshold τ, we consider the reported location as *anomalous*, i.e., not consistent with the observed profile. In such a case, additional countermeasures can be taken, such as warning the drone, jamming it or physically shutting it down. Otherwise, the location reported is considered *genuine*, and no further action is taken. Note that our methodology is customizable and further extensible with the desired solutions for range estimation, location estimation, and location evaluation. The next sections provide details on the algorithms we used for range estimation (Sect. V-B), location estimation (Sect. V-C) and location evaluation (Sect. V-D).

B. Range Estimation at the Observer

Let us denote with $r_{k,n}$ the RSS of the n-th packet received at the observer k. Range estimation techniques allow to obtain an estimated distance $\bar{d}_k$ from $r_{k,n}$, using a function F. Such function depends on the specific wireless propagation model adopted to model the wireless channel. For this work, in line with the relevant literature on propagation models for UA communications [20], [21], we adopt the Log-Distance Path Loss (LDPL) model (we will also validate this choice through real data in Sect. VI-A). The LDPL model formalizes the path loss at a given distance from the transmitter ($PL(d_k)$) as a logarithmic function, as in Eq. 1.

$$PL(d_k) = P_T - r_{k,n} = PL(d_0) + 10\gamma \log_{10}(\frac{d_k}{d_0}) + X_g(0, \sigma^2), \tag{1}$$

where $PL(d_0)$ is the path-loss at a reference distance d_0 from the transmitter, γ is the path-loss exponent, and $X_g(0, \sigma^2)$ is a Gaussian random variable with zero mean and variance σ^2 that models the effects of shadowing and multipath fading [22]. We can rework Eq. 1 to obtain the distance $d_{k,n}$ of the k-th receiver from the transmitter for the n-th packet, according to Eq. 2.

$$d_{k,n} = d_0 \cdot 10^{\frac{(P_T - r_{k,n} - PL(d_0))}{10\gamma}}. \tag{2}$$

Each observer applies Eq. 2 for each received packet, obtaining a vector of estimated distances, namely $\hat{\mathbf{d_k}}$. We denote with $\mathbf{l_{k,n}}$ the location reported by the drone in the n-th RID message received at the k-th observer, in terms of latitude, longitude, and altitude, and with all distances $\mathbf{L_k}$ reported by the drone at the k-th observer in the time frame T. Each observer forwards to the central server the estimated distances $\hat{\mathbf{d_k}}$ and the reported locations $\mathbf{L_k}$.

C. Location Estimation

The location estimation building block aims to combine the various range measurements of the observers to obtain a vector of estimated locations of the UA, namely $\hat{L}_k$. We first apply an outlier removal process on the range measurements to filter out anomalous samples, i.e., estimated distance values that, due to anomalous noise, are far away from other values. To this aim, for each n-th packet, we generate the set of all possible combinations of s distinct observers, denoted as $\{C_a \mid 1 \leq a \leq A\}$, where A is the overall number of combinations of s observers out of K, i.e., $A = \binom{s}{K}$. For each combination C_a, we perform multi-lateration using Nonlinear Least Squared Error (NLSE), according to Eq. 3.

$$\hat{L}_{(n,a)} = \min_{e} \sum_{s_i=1}^{s} \left\| \hat{d_{s_i,n}} - d_{s_i,n} + e_n \right\|^2, \tag{3}$$

where $\|\cdot\|$ defines the norm-2 operator and $e_n \in X_g(0, \sigma^2)$ (recall Eq. 1). For each combination C_a of s observers ($s \leq K$) and n-th packet, such a process yields an estimated location $\hat{L}_{(n,a)} = (\hat{x}_{(n,a)}, \hat{y}_{(n,a)}, \hat{z}_{(n,a)})$. Subsequently, we derive the estimated location from the n-th packet by computing a combined estimate $\hat{L}_n = (\hat{x}_n, \hat{y}_n, \hat{z}_n)$ over all combinations, by computing the statistical median of the corresponding coordinates, as in Eq. 4.

$$\begin{aligned} \hat{x}_n &= \text{Median}_{a=1:A}(\{\hat{x}_{(n,a)}\}), \\ \hat{y}_n &= \text{Median}_{a=1:A}(\{\hat{y}_{(n,a)}\}), \\ \hat{z}_n &= \text{Median}_{a=1:A}(\{\hat{z}_{(n,a)}\}). \end{aligned} \tag{4}$$

We denote $\hat{L}_n = (\hat{x}_n, \hat{y}_n, \hat{z}_n)$ as the vector of *estimated locations*.

D. Location Evaluation

In this phase, we evaluate the location reported by the drone, with the aim of denoting it as *genuine* or *anomalous*. Specifically, for each n-th message delivered by the drone, we compute the Euclidean distance element-wise, i.e., between each element in the vector of the reported locations ($\mathbf{L_n}$) and each element in the vector of estimated locations($\hat{L}_n$), and we finally take the median value of the resulting vector, namely δ, as in Eq. 5.

$$\delta = \text{Median}_{n=1:N}\{\sqrt{(x_n - \hat{x}_n)^2 + (y_n - \hat{y}_n)^2 + (z_n - \hat{z}_n)^2}\}. \tag{5}$$

We denote δ as the *decision value*. Note that, for the case of a single observer ($K = 1$), we can only compute ranges and not estimated locations. Therefore, in this case, the observer computes the absolute difference between the distance claimed in the k-th packet ($d_{(n,k)}$) and the estimated distance ($\hat{d}_{(n,k)}$), and then takes the median of all the packets as the decision value, δ, as per Eq. 6.

$$\delta = \text{Median}_{n=1:N}(\{|\hat{d}_{(n,k)} - d_{(n,k)}|\}) \tag{6}$$

We finally compare the decision value δ to the decision threshold τ, calibrated during training. If $\delta \leq \tau$, the location(s) reported by the drone is (are) *genuine*. Otherwise, if $\delta > \tau$, the location(s) reported by the drone is (are) *anomalous*. In such cases, many countermeasures can be triggered, e.g., warning the drone, jamming the communication link with the controller or the GPS to make the drone land or go back, and finally physically taking the drone down. The best approach depends on the specific situation and is beyond the scope of this work.

VI. Performance Evaluation

A. Reference Dataset

To validate our solution, we used the real data provided as part of the ICMCIS dataset, available publicly at [23]. The dataset was created specifically for a competition held in September 2020 at the International Conference on Military Communications and Information Systems (ICMCIS). The primary objective of the competition was to develop predictive models and algorithms to accurately predict the future location of a drone based on its past movements. The dataset comprises log files obtained from multiple UAs following predetermined flight paths, in an area of $1.5km \times 2.5km$ [24]. The designated testing site is a field at Luitenant-general Bestkazerne in De Peel, Netherlands. Each UA log file provides data points at a granularity of 100 ms, including a UTC timestamp (in ms), the current location of the UA (latitude, longitude, altitude) and instantaneous velocity. The UAs operated within the coverage range of two distinct radar systems and two RF Direction Finding (DF) systems, whose log files are also available. For this research, we use the log files of one of the RF DF systems, i.e., the one with the fictional name *Diana*. The log files of the Diana sensor provide, for each received packet from three UAs, a UTC timestamp, a target identifier, the Signal-to-Noise Rate (SNR) in decibels, and the source of the signal, denoted either as *controller* or *aircraft*. As the time on the UAs and the sensor are synchronized, we can assume that the location of the UA is delivered wirelessly to the sensor, as in a real RID scenario. We used the above-described data to experimentally model the actual channel experienced by an observer. Then, we used the derived experimental propagation model to run both simulations (Sect. VI-B) and real data analyses (Sect. VI-C). Finally, note that we do not consider packet losses.

TABLE II
Fitting of FSPL, LDPL, and TRGR models to the data in the dataset at [23], in terms of RMSE, MAE and R^2 metrics.

Propagation Model	**Statistic**	**UA 1**	**UA 2**	**UA 3**
FSPL	$RMSE$	4.79	6.81	6.01
	MAE	3.47	5.57	4.83
	R^2	0.62	-0.28	-0.43
LDPL	$RMSE$	3.45	4.04	3.54
	MAE	2.36	2.96	2.59
	R^2	0.8	0.55	0.5
TRGR	$RMSE$	11.73	14.56	14.69
	MAE	9.82	12.24	12.09
	R^2	-1.29	-4.83	-7.56

Channel Estimation. As a first step, we investigated the most suitable propagation model, i.e., the one that best fits the real data. To this end, we first converted the SNR values to RSS values by subtracting the noise floor level, which was experimentally verified to be approximately 95 dBm. We selected three models for evaluation, according to relevant literature on air-to-ground propagation modelling [20], [25], [22], i.e., the LDPL model, the Free-Space Path Loss (FSPL), and the Two-Ray Ground Reflection (TRGR). Then, we fit the three candidate models to the empirical data available for the three UA, using the NLSE method as implemented by the tool *curve_fit()* of the SciPy Python library [26], and we evaluated the goodness of fit of each model to the empirical data using various statistical measures. Specifically, we used well-known metrics for the evaluation of the goodness of fit of a model to real-valued data, i.e., the Root Mean-Squared Error (RMSE), Mean Absolute Error (MAE), and R^2 metrics [27]. We summarize the results of our analysis in Tab. II. For all the UAs, the LDPL model is the one reporting the lowest values of

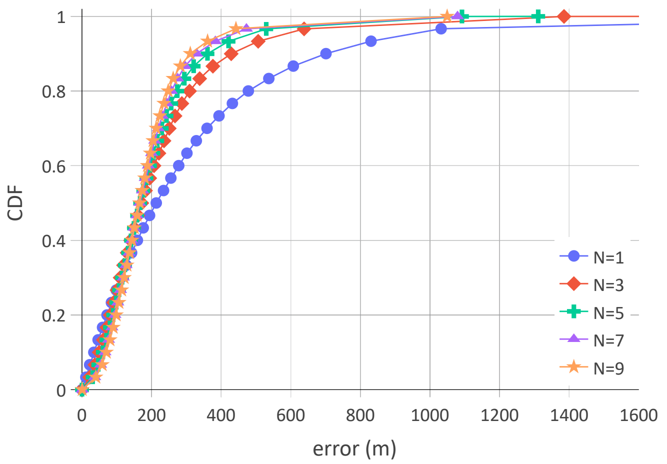

Fig. 3. CDF of the localization error with $K = 1$ observer, increasing RID packets.

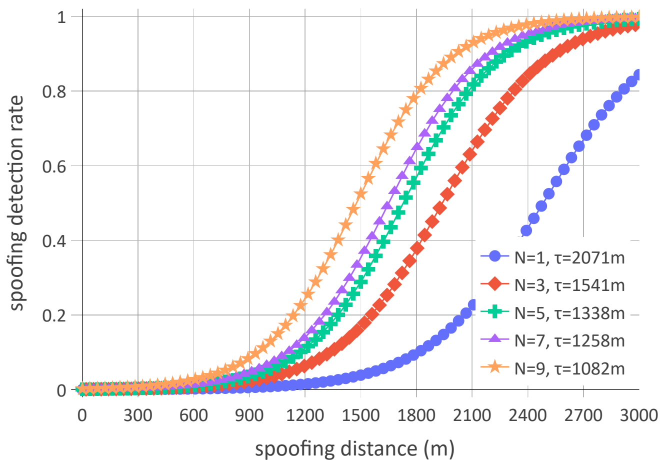

Fig. 4. Spoofing detection rate with $K = 1$ observer, increasing RID packets.

RMSE and MAE, as well as the highest values of R^2, thereby being the best fit to our data. Finally, we applied the Least-Squares method to estimate the values of the parameters of the LDPL model for the three UAs, as reported in Tab. III.

TABLE III
ESTIMATED PARAMETERS OF THE LDPL MODEL FROM REAL DATA.

UA	γ	d_0 **[m]**	$PL(d_0)$ **[dB]**
UA 1	1.35	1	58.07
UA 2	0.9	1	70.82
UA 3	0.84	1	69.88

B. *Simulation Analysis*

We start our analysis by evaluating the performance of our solution under controlled conditions, using the model (LDPL) and parameters obtained from real data. We investigated two scenarios, i.e., a single observer (General Public Observer) and multiple observers (Public Safety Observers). We implemented our solution using Python v3.9 and, for each scenario and configuration, we performed $1,000$ runs and reported the results using the Cumulative Distribution Function (CDF), to show the statistical distribution of the results.

Single Observer Setup. We first focus on the effect of an increasing number of RID packets on the overall localization accuracy and its capability to detect drones spoofing their reported locations. Fig. 3 reports the CDF of the localization error (in meters) for the case of a single receiver and $\sigma = 3.5$ dB, considering various RID packets ($N \in [1, 9]$)). For each of the CDFs, we picked the distance corresponding to the CDF value 1, to have 0 False Negatives, and used it as the threshold τ of our solution. Figure 4 reports the spoofing detection rate of our solution where, starting from a random location, we let the reported locations of the UA drift away from the actual one with increasing spoofing distance $\tilde{d}$. For all the results below, the legend in the figures also reports the value of the threshold τ. Overall, increasing the number of used RID packets enhances the spoofing detection capability of the observer. With 3 packets, the system can detect 95 % of the attacks only when the drone spoofs its location at a distance of $2,452$ m from the actual one. Consequently, such a spoofing detection rate is achieved at a distance of $2,351$ m with 5 packets and $2,067$ m with 9 packets. The price to pay for such a performance gain is the time to take a decision, which depends on the overall time to receive such packets. We also investigated the impact of different channel conditions. We adopted the same methodology as before for the spoofing experiment, i.e., for each of the CDFs obtained for the localization error, we picked the distance corresponding to the CDF value 1, and we used it as the threshold τ of our solution. Below, due to the limited available space, we report only the results of the location spoofing detection rate in the considered scenarios. We report the results in Fig. 5. The results demonstrate that channel conditions impact the capability to detect location spoofing attacks. With the most challenging conditions $\sigma = 4.5$ dB, we achieve a spoofing

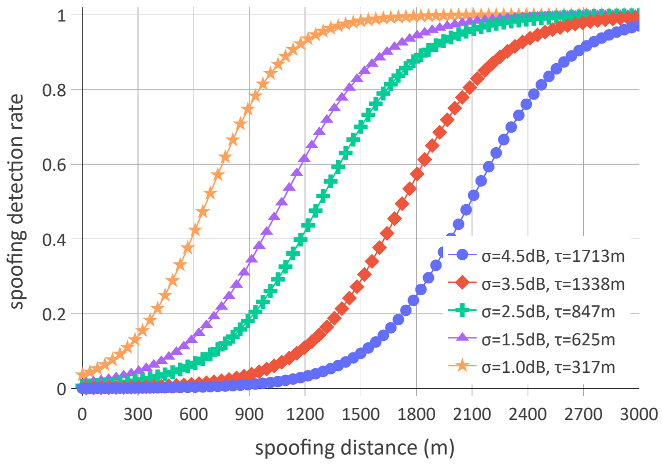

Fig. 5. Spoofing detection rate with a single observer, with various channel conditions.

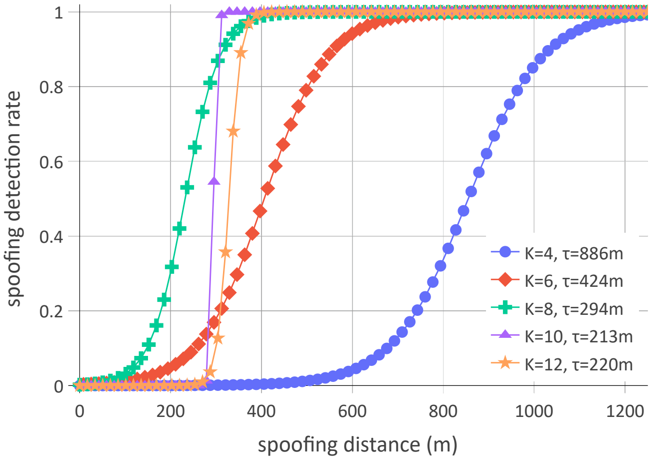

Fig. 6. Spoofing detection rate with with multiple observers, $N = 5$ packets and $\sigma = 3.5$ dB.

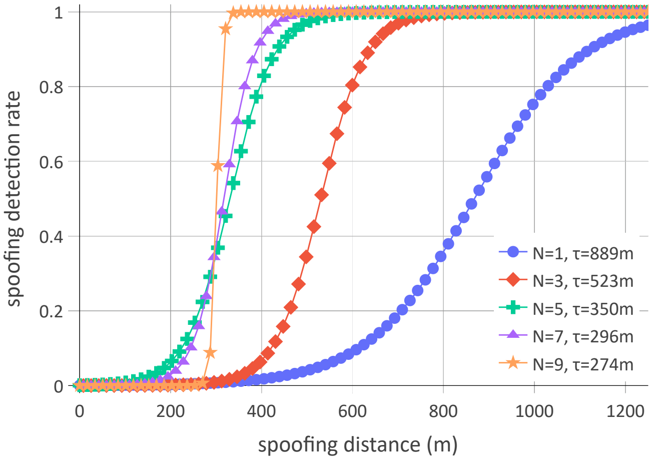

Fig. 8. Spoofing detection rate with $K = 8$ observers, $\sigma = 3.5$ dB and various number of packets.

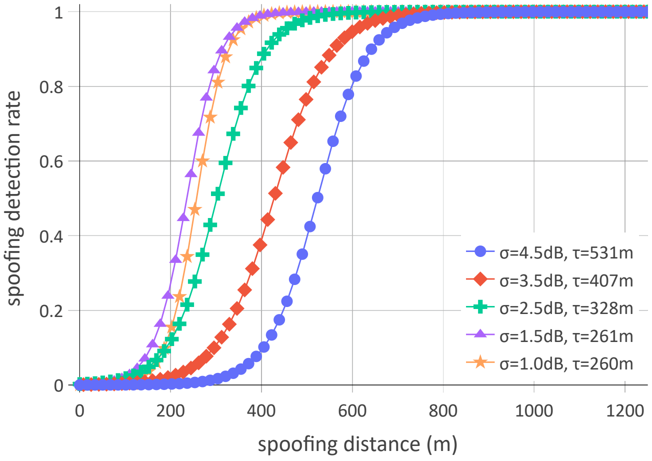

Fig. 7. Spoofing detection rate with $K = 8$ observers, $N = 5$ packets and various channel conditions.

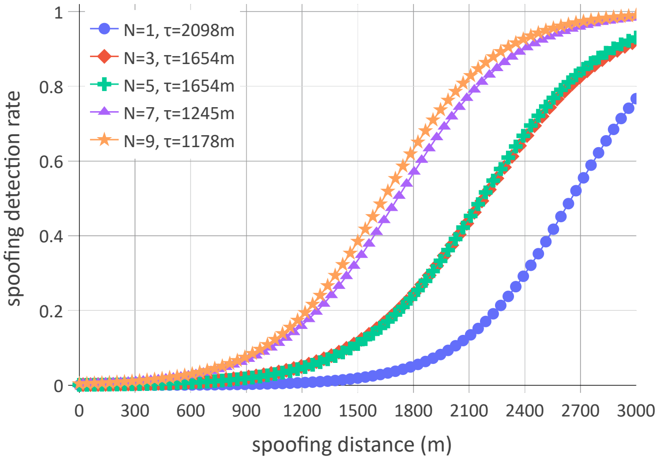

Fig. 9. Spoofing detection rate with real data from *Diana* sensor, UA 1.

detection rate of 0.95 with a spoofing distance of $2,797$ m, while such value reduces to $1,337$ m with better conditions ($\sigma = 1$ dB). For both experiments, the accuracy of the system is severely limited by the availability of a single observer. We can overcome such an issue using multiple observers.

Multiple Observers Setup. We first considered $N = 5$ packets and $\sigma = 3.5$ dB, and we varied the number of deployed observers. Fig. 6 summarizes the results of our investigation. The results show the positive effect of multiple observers on the spoofing detection rate. With $K = 4$ observers, the system detects 95% of the spoofing attacks at a spoofing distance of $1,199$ meters, while the same performance is achieved for a spoofing distance of only 270 m with $K = 12$ observers. For completeness, we also report below the spoofing detection rate achieved by our solution with $K = 8$ observers, with various channel conditions (considering $N = 5$ packets) and an increasing number of packets (considering $\sigma = 3.5$ dB) in Fig. 7 and Fig. 8, respectively.

C. Real Data Analysis

Following our previous analysis, we tested our solution on real data from the sensor *Diana* of the ICMCIS dataset.

Single Observer Setup. We use the parameters estimated empirically in Sec. VI-A, and we considered multiple RID packets, for all the three UAs in the dataset. Figs. 9, 10, and 11 show the results of our analysis. The performances are very similar to the ones obtained via simulations for $\sigma \approx 4.5$ dB. With 9 RID packets, to mention some reference results, we can detect with 95% probability drones spoofing their location at a distance of $2,513$ meters for UA 1, and $2,675$ meters for UA 3, while the value for UA 2 is greater than 3 km. Such large values are due to both the single receiver setup and to the experienced channel conditions, as shown in Sect. VI-B.

Multiple Observer Setup. We extended our analysis to the scenario of multiple observers. Since the reference dataset provides data for one sensor and multiple UAs, we used such real data to model the channel experienced by multiple

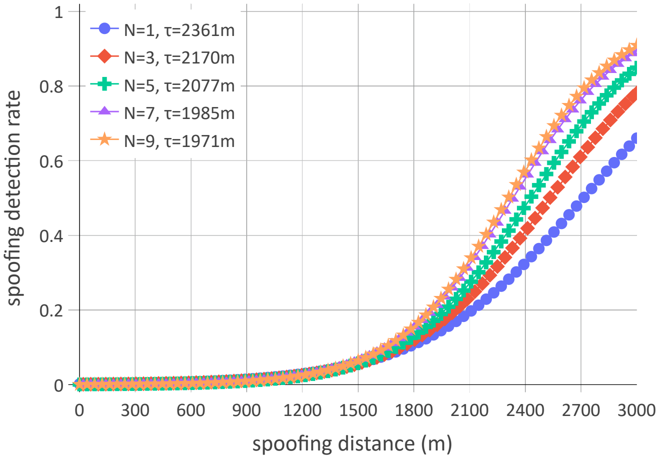

Fig. 10. Spoofing detection rate with real data from *Diana* sensor, UA 2.

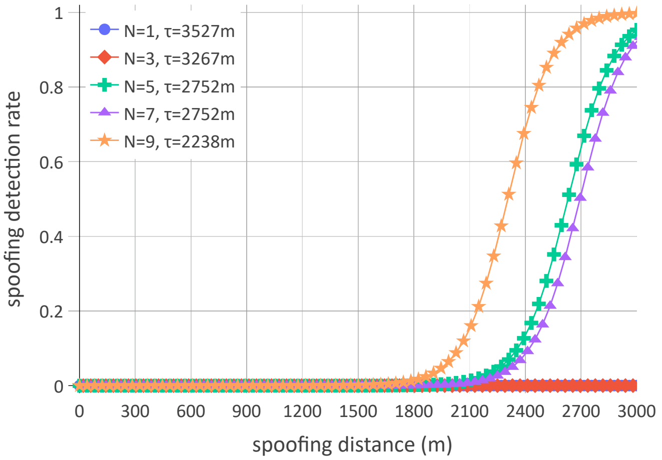

Fig. 11. Spoofing detection rate with real data from *Diana* sensor, UA 3.

observers, randomly placed in the area. In Figs. 12, 13, and 14, we report the spoofing detection rate for various numbers of observers considering each of the three UAs individually and $N = 5$ RID packets. In line with the previous results, the worst performance is obtained for the UA 3 where, with $K = 12$ observers, we can detect 95% of the spoofing location reports a distance of $1,175$ m from the actual one. This value decreases to 364 m for UA 1 with $K = 12$. Note that our results are very similar to those obtained by simulations (Fig. 7) for $\sigma \approx 4.5$ dB. Thus, our performance is significantly affected by the experienced channel conditions and shows the potential for significant improvement with less noisy data.

D. Discussion

Impact of RSS quantization In our analysis, we used RSS values truncated at their integer value, without decimal digits. This is reasonable when considering generic Commercial Off-The-Shelf (COTS) receivers. However, CIs with a larger budget might opt to use Software-Defined Radios (SDRs), able to log RSS values with the desired accuracy. However,

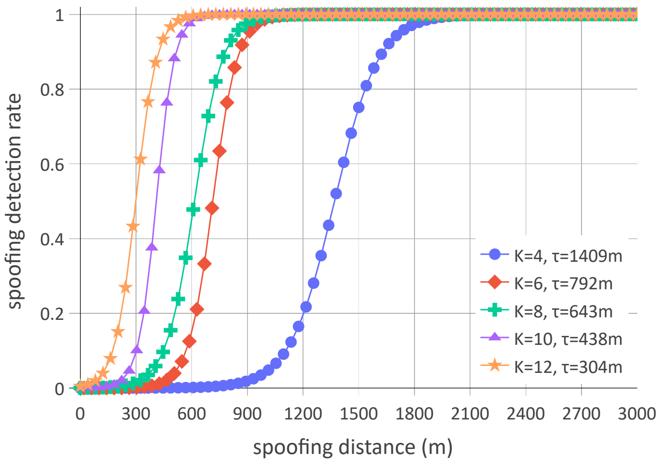

Fig. 12. Spoofing detection rate with multiple observers and real data of the channel experienced for UA 1.

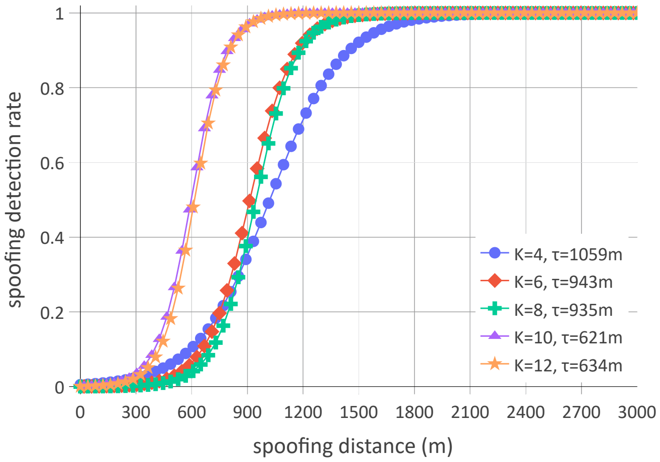

Fig. 13. Spoofing detection rate with multiple observers and real data of the channel experienced for UA 2.

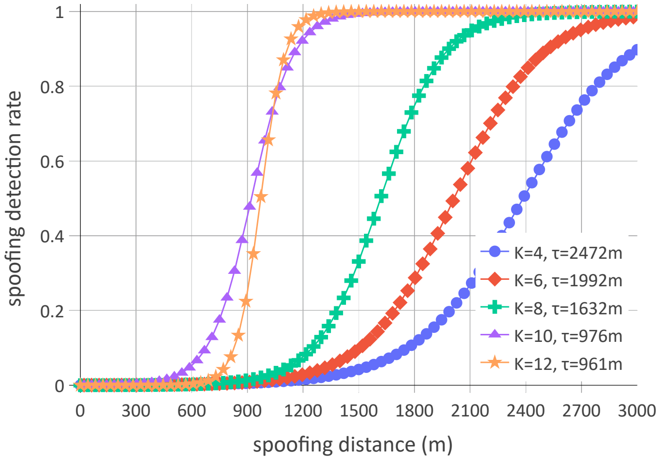

Fig. 14. Spoofing detection rate with multiple observers and real data of the channel experienced for UA 3.

using such receivers does not always bring benefits to the CI operators. In fact, we verified that the use of more accurate observers improves the spoofing detection rate only when the level of noise that affects the wireless channel is low. Instead, when the channel is noisy, they do not enhance performance.

Manipulating the Transmission Power Level. If the adversary ϵ knows the location of an observer, it can rework the LDPL model to obtain the transmission power to use to let the observer estimate the desired (fake) location, as in Eq. 7.

$$\bar{P_T} = P_T + 10\gamma * \log_{10}(d_k/\bar{d_k}), \qquad (7)$$

where $\bar{P_T}$ is the transmission power required to deceive the observer, P_T is the nominal (legitimate) transmission power of the UA, d_k is the actual distance between the UAV and the k-th observer, γ is the path loss exponent, and $\bar{d_k}$ is the deceiving distance, i.e., the distance between the fake location that the drone wants to report and the location of the n-th observer. However, we first notice that the application of such an attack requires knowledge of the location of the observer, which is not trivial. Also, with multiple observers, assuming a drone using an omnidirectional antenna, Eq. 7 should be satisfied for each deployed observer distance d_k. Thus, ϵ should solve a system of K equations and one unknown variable, which is unsolvable by definition. The best ϵ can do is setting P_T to deceive one observer, while the others will estimate another location, different from both the real and the fake (desired) one. Since ϵ has no control over the process, there are overwhelming chances that the difference between the location reported in the RID message and the estimated one exceeds τ, leading to detection. Alternatively, ϵ might use multiple directional antennas, each transmitting crafted messages to specific observers at known positions. However, bringing multiple directional antennas on board a drone requires space and power from the drone, and the probability that ϵ deceives the system still depends on the mutual displacement among observers, making the attack highly unreliable.

VII. Conclusion and Future Work

In this paper, we have investigated the feasibility of using wireless localization techniques to detect drones that perform location spoofing attacks via RID messages. To this end, we designed GhostBuster, a modular solution for the detection of malicious drones that integrates a RSS-based localization technique. We validated our solution using both simulations and data from real drone flights. We experimentally demonstrated that systems comprising multiple receivers can detect finer location spoofing attacks. At the same time, channel conditions significantly affect the effectiveness of the system. Overall, our research shows that wireless localization solutions, if carefully deployed, can successfully detect drones falsifying the reported location, taking advantage of currently-enforced RID regulations. In the future, we plan to extend GhostBuster by integrating techniques based on the Time of Arrival.

Acknowledgements

This work has been partially supported by the INTERSECT project, Grant No. NWA.1162.18.301, funded by the Netherlands Organisation for Scientific Research (NWO). The findings reported herein are solely responsibility of the authors.

References

[1] K. Belwafi, et al., "Unmanned Aerial Vehicles' Remote Identification: A Tutorial and Survey," *IEEE Access*, vol. 10, pp. 87 577–87 601, 2022.

[2] FAA, "UAS Remote Identification Overview," https://www.faa.gov/uas/getting_started/remote_id/, 2021.

[3] European Union Aviation Safety Agency, "Commision Delegated Regulations (EU) 2020/1058," https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=celex\%3A32020R1058, 2020.

[4] DeDrone, "Worldwide Drone Incidents," https://www.dedrone.com/resources/incidents-new/all, 2023, (Accessed: 2023-Jun-23).

[5] S. Park et al., "Survey on Anti-Drone Systems: Components, Designs, and Challenges," *IEEE Access*, vol. 9, pp. 42 635–42 659, 2021.

[6] UASWeekly, "Hidden Level's Drone Tracking System Ready-Made for New FAA Rules," https://uasweekly.com/2022/12/12/hidden-levels-drone-tracking-system-ready-made-for-new-faa-rules/, 2023, (Accessed: 2023-Jun-23).

[7] L. Oliveira, et al., "Mobile Localization Techniques for Wireless Sensor Networks: Survey and Recommendations," *ACM Trans. on Sensor Netw.*, vol. 19, no. 2, pp. 1–39, 2023.

[8] F. Ricciato, et al., "Position and Velocity Estimation of a Non-Cooperative Source From Asynchronous Packet Arrival Time Measurements," *IEEE Trans. on Mob. Comput.*, vol. 17, no. 9, 2018.

[9] H. Chen, et al., "Improved Robust TOA-based Localization via NLOS Balancing Parameter Estimation," *IEEE Trans. on Veh. Technol.*, vol. 68, no. 6, pp. 6177–6181, 2019.

[10] J. Ying, et al., "Precision of RSS-based Localization in the IoT," *Int. Journ. of Wirel. Informat. Netw.*, vol. 26, no. 1, pp. 10–23, 2019.

[11] B. Mukhopadhyay, et al., "RSS-based Localization in the Presence of Malicious Nodes in Sensor Networks," *IEEE Trans. on Instrument. and Measurem.*, vol. 70, pp. 1–16, 2021.

[12] M. Monteiro, et al., "Detecting malicious ADS-B broadcasts using wide area multilateration," in *IEEE/AIAA Digital Avionics Systems Conference*, 2015, pp. 4A3–1–4A3–12.

[13] N. Ghose, et al., "Verifying ADS-B navigation information through Doppler shift measurements," in *IEEE/AIAA Digital Avionics Systems Conference*, 2015, pp. 4A2–1–4A2–11.

[14] A. Darabseh, et al., "MAVPro: ADS-B Message Verification for Aviation Security with Minimal Numbers of on-Ground Sensors," in *ACM Conf. on Security and Privacy in Wirel. and Mob. Netw.*, 2020, p. 53–64.

[15] Y. Kim, et al., "A secure location verification method for ADS-B," in *IEEE/AIAA Digital Avionics Systems Conference*, 2016, pp. 1–10.

[16] E. Wisse, et al., "A^2RID-Anonymous Direct Authentication and Remote Identification of Commercial Drones," *IEEE Internet of Things J.*, 2023.

[17] Electronic Code of Federal Regulations, "14 CFR § 89.310 - Minimum performance requirements for standard remote identification unmanned aircraft," https://www.law.cornell.edu/cfr/text/14/89.310, 2023, (Accessed: 2023-Jun-23).

[18] D. Micault, M. Boucadair, "Drone Remote Identification Protocol (DRIP) Charter - charter-ietf-drip-01," IETF, Tech. Rep., 2022.

[19] S. Card, et al., "Drone Remote Identification Protocol (DRIP) Architecture - draft-ietf-drip-arch-31," IETF, Tech. Rep., 2023.

[20] W. Khawaja,et al., "A Survey of Air-to-Ground Propagation Channel Modeling for Unmanned Aerial Vehicles," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2361–2391, 2019.

[21] J. Parsons, et al., *The Mobile Radio Propagation Channel*. Wiley New York, 2000, vol. 2.

[22] A. Goldsmith, *Wireless communications*. Cambridge university press, 2005.

[23] NATO, "Drone identification and tracking," https://www.kaggle.com/c/icmcis-drone-tracking/overview, 2021, accessed: 2023-March-07).

[24] A. Brighente, et al., "Hide and Seek: Privacy-Preserving and FAA-compliant Drones Location Tracing," in *Int. Conf. on Availability, Reliability and Security*, 2022, pp. 1–11.

[25] E. Vinogradov, et al., "Tutorial on UAV: A blue sky view on wireless communication," *Jour. of Mob. Multimedia*, vol. 14, no. 4, pp. 395–468, 2019.

[26] P. Virtanen, et al., "SciPy 1.0: fundamental algorithms for scientific computing in Python," *Nature methods*, vol. 17, no. 3, pp. 261–272, 2020.

[27] C. Sammut and G. I. Webb, “Mean Absolute Error,” *Encyclopedia of Machine Learning*, vol. 652, 2010.