# #PrettyFlyForAWiFi : Real-World Detection of Privacy Invasion Attacks by Drones


SIMON BIRNBACH, University of Oxford, United Kingdom
RICHARD BAKER, University of Oxford, United Kingdom
SIMON EBERZ, University of Oxford, United Kingdom
IVAN MARTINOVIC, University of Oxford, United Kingdom



Drones are becoming increasingly popular for hobbyists and recreational use. But with this surge in popularity comes increased risk to privacy as the technology makes it easy to spy on people in otherwise-private environments, such as an individual's home. An attacker can fly a drone over fences and walls in order to observe the inside of a house, without having physical access. Existing drone detection systems require specialist hardware and expensive deployment efforts, making them inaccessible to the general public.

In this work we present a drone detection system that requires minimal prior configuration and uses inexpensive commercial off-the-shelf (COTS) hardware to detect drones that are carrying out privacy invasion attacks. We use a model of the attack structure to derive statistical metrics for movement and proximity that are then applied to received communications between a drone and its controller. We test our system in real world experiments with two popular consumer drone models mounting privacy invasion attacks using a range of flight patterns. We are able both to detect the presence of a drone and to identify which phase of the privacy attack was in progress while being resistant to false positives from other mobile transmitters. For line-of-sight approaches using our kurtosis-based method, we are able to detect all drones at a distance of 6 meters, with the majority of approaches detected at 25 meters or further from the target window without suffering false positives for stationary or mobile non-drone transmitters.



CCS Concepts: • **Security and privacy** → **Systems security**; **Mobile and wireless security**.

Additional Key Words and Phrases: uav, drones, drone detection, privacy invasion

**ACM Reference Format:**
Simon Birnbach, Richard Baker, Simon Eberz, and Ivan Martinovic. 2020. #PrettyFlyForAWiFi : Real-World Detection of Privacy Invasion Attacks by Drones. *ACM Trans. Priv. Sec.* 37, 4, Article 111 (August 2020), 35 pages. https://doi.org/10.1145/1122445.1122456


## 1 INTRODUCTION

### 1.1 Motivation

Drones have become enormously popular in recent years. Since the release of the Parrot AR.Drone in 2010, examples of the technology have exploded into the public consciousness. The use of drones, or variously unmanned aerial vehicles (UAVs) or remotely-piloted aerial systems (RPAS), is no longer restricted to military and professional domains. It has opened up to laymen as well. Self-stabilizing multirotor designs that automatically hover upright have made piloting drones easily accessible for newcomers. Meanwhile, advances in sensor design and image processing have brought


Authors' addresses: S. Birnbach, R. Baker, S. Eberz, and I. Martinovic, University of Oxford, Department of Computer Science, Wolfson Building, Parks Road, Oxford OX1 3QD, United Kingdom; emails: {simon.birnbach, richard.baker, simon.eberz, ivan.martinovic}@cs.ox.ac.uk.

(a) View into window at distance of about 10m. The FOV is narrow and the reflections mask view. Little of the room is visible.

(b) View into window at close-range of roughly 2m. The FOV is wider and the reflections less dominant. The door, curtains, red walls and bed can be seen.

Fig. 1. Examples of effect of distance on surveillance footage quality.

sophisticated flight-assist features to further ease the burden. Modern consumer drones can be flown using a handheld controller and a smartphone; and while safety records do vary wildly it is completely achievable for a novice to operate this type of aircraft without incident despite no prior training whatsoever.

The opportunity to take to the skies without arduous preparation has clearly appealed to many people. Industry analyst firms have reported sales of drones increasing every year since 2014, with the global drone market valued at an estimated $4.9 billion for 2019 [16, 18]. The US Federal Aviation Administration reported in January 2018 that there were over a million drones on their national register [2].

Sales by the market-dominant DJI Technology were reported at 375,000 units in 2017, with the majority intended for photography [36]. Indeed, photography is one of the most popular uses and the inclusion of high-quality, onboard camera equipment is a recognized selling point for a device.

However, the rise of consumer drones has brought many problems with it. The use of drones to deliver weapons, drugs and other contraband into prisons is now commonplace [5], political figures have been targeted in drone-based protests [4, 34] and the repeated appearance of a drone closed Gatwick airport for two days in December 2018 [13]. More localized abuses are often reported as well and there has been an increasing unease in the general population about privacy invasion by drones carrying high-fidelity camera equipment. As most consumer models are outfitted with cameras for live-view video during flight, drones can fly over fences to see into nearby gardens or even look into windows to observe the interior. In Seattle, a woman called the police after spotting a drone flying in front of her 26th-floor apartment window; observing her partially-undressed inside [30]. In a famous landmark case, a man from Kentucky was arrested after shooting down a drone over his property. He explained this act by saying that the drone was spying on his sunbathing daughter [38].

While systems for detecting drones are available, they are specialized and costly; intended for law enforcement use to protect secure sites and VIPs, rather than for normal citizens. By contrast, we focus on these threats to private individuals. An attacker can launch a drone from a distant location and fly it to an otherwise-inaccessible private area, such as outside a top-floor window within a walled garden, and freely observe the occupants. To launch such an attack the attacker has to move close enough to the window to get a full view of the inside of the house. This view of the interior depends on the field-of-view (FOV) of the window, as Figure 1 demonstrates. As a window is a distant aperture of fixed size, the FOV initially increases very slowly during the approach before expanding rapidly once the drone is at close range. The occupants require a system that alerts them to the presence of an impending privacy invasion, with sufficient warning to allow them to react accordingly and before the drone gets close enough to allow observation. The system must be easy to deploy without specialist knowledge and detect a wide variety of drones based on intrinsic properties of UAV activity; both to be robust to the rapid pace of advancement in drone technology and to avoid spurious alarms. To be most readily-applicable, the system should avoid additional hardware requirements and take advantage of existing sensing capabilities.

In this paper, we present a system that detects the presence of a live-streaming UAV that is being used to invade someone's privacy in their home. We make use of readily-available commercial, off-the-shelf (COTS) hardware to measure variations in the strength of the drone's radio control signal and develop metrics that identify signal properties inherent to drone flight. By using a radio-based approach, the system is equally effective in light or dark conditions and as the signal measurements are commonly provided by radio chipsets, the system also has the potential to be a software-only change to existing in-home devices. In developing the system, we make the following specific contributions:

- Development of a model of UAV-based privacy-invasion attacks
- Derivation of statistical metrics to identify drone movement and proximity
- Proposal of mechanisms to overcome attempts by an attacker to avoid detection, such as varying their speed or flight pattern
- Evaluation of the system in a real-world scenario using popular consumer drones and various moving benign transmitters

### 1.2 Related work

In this section, we will first survey existing UAV detection methods and subsequently give a short overview of available defense mechanisms.

There are a range of methods that can be used to detect drones. Research efforts have been dedicated to improving techniques for each method, although all have notable limitations and many are costly and difficult to operate. Commercial systems generally adopt hybrid methods that combine several techniques together, although these only exacerbate cost problems further. We describe below key results that show the behavior of different methods, a more in-depth survey is presented in [24]. Many of the surveyed papers provide no performance metrics and those that do often rely on different measures than others, making direct comparisons challenging. Nevertheless, we present performance metrics where available.

The most traditional approach is through the use of radar; a staple of military and aircraft control for a long time. However, the design of radar systems (particularly the frequency of transmitted radio waves) must be optimized for a given aircraft size, so radars used for conventional aircraft monitoring cannot track objects as small as drones. Instead, high-frequency radars well into the GHz range are required [12, 17]. While such systems have been shown to be able to distinguish birds, drones and small planes with a probability of correct classification of the order of 95% in an ideal setting [20], they are prone to exhibit performance degradation in adverse weather conditions. More importantly, deploying any radar system is an undertaking that is out of reach for normal individuals; requiring expensive equipment, transmission licensing and trained operators.

Image and video analysis can be used to detect UAVs by using both their appearance and motion cues, with an average precision of 0.751 quoted in [28], although the growing variation in drone shapes is challenging for appearance-based approaches and methods based on motion cues struggle with similarities between drone and bird movement [9]. More generally, visual detection requires line-of-sight and is affected by light levels and visibility conditions. The use of infra-red emissions instead of, or as well as, visible light improves performance in darkness, but still suffers from confusion of drones and birds as well as reduced performance where background temperature levels are similar to drone surface temperatures [7]. Irrespective of the type of light observed, real-time image processing is computationally expensive and not readily incorporated into existing consumer devices.

LIDAR systems have shown recent promising results in accurate detection, but again require line-of-sight conditions and are a far more expensive technology than conventional cameras [11].

Acoustic detection, using microphone arrays or acoustically-augmented cameras, circumvents many of the limitations of visual methods by being effective at night or with no direct line-of-sight [9], achieving a 99.5% probability of detection with a false positive rate of 3% for ranges smaller than 600m in good conditions [6]. However, such systems suffer from acoustic variants of the same problems; namely that sound distortion in bad weather or noisy environments can limit performance. Systems also rely upon known acoustic signatures in order to detect drones and must be kept up to date. While acoustic systems can be built with off-the-shelf equipment, even comparatively cheap examples are still beyond the means of most private individuals. The low-cost acoustic array system in [10] requires an array of 24 microphones and cost $3768 to build.

Hybrid approaches such as CSUAV [37] and ADS-ZJU [32] combine radar or radio frequency (RF), acoustic arrays and video cameras to profit from the benefits of all approaches. The authors of [32] report individual detection probabilities of 76.9%, 98.1% and 88.2% for the audio, video and RF submodules, respectively, when the drone is at a distance of 50m. The corresponding false alarm probabilities are 2.2%, 1.1% and 2.8%. But while the combined use of these approaches improves the detection probability to 99.3% with a false alarm probability of 5%, they also increase the complexity and cost of the system substantially. It is also noteworthy that the authors attribute the low false alarm probability of the video submodule to a lack of occlusions during the experiments. In contrast, we have designed our experiment in a way that considers likely sources of false positives and explicitly tests the system against them.

Boddhu et al. [8] propose an approach that uses humans as sensors by building a collaborative smartphone app that allows users to share drone sightings. Their approach is more appropriate to target large-scale threats, and is unlikely to be helpful for the defense of a single property as no reliance can be made on the general public to provide a report.

We focus instead on methods that use RF signal processing for detection. Some RF-analysis based methods use localization to detect drones simply as flying transmitters. However, these methods generally require multiple receivers and costly, precise time-synchronization mechanisms between the receivers [29]. As our work is intended for use by private individuals, we focus on the use of inexpensive consumer technologies, and the reuse of existing domestic communication systems to permit easily-accessible drone detection. Early academic work in this area has used protocol signatures from the drone's Wi-Fi connection and MAC address prefixes to recognize drones [26]. These methods depend upon known manufacturer lists and unencrypted communications however. A contrasting example is work on the use of low-level physical effects, observed in an RF signal, to identify drones [25]. This approach is able to recognize drone transmitters based on movement characteristics with an accuracy of 96.5%, a recall of 97% and a precision of 95.9% at a distance of 10m, but it requires access to the raw radio signal. This in turn requires either the deployment of a purpose-built detector or the use of software-defined radio implementations, which make it expensive compared to a system using only commonplace consumer devices. Crucially, in both prior cases, the systems are also unable to distinguish a neighbour turning on a drone in their house, or a nearby flight with no malicious intent, from an actual privacy-invasion attack. More recent systems have observed higher-level metrics and used a visual stimulus at a potential target site in order to inject a controlled signal into the live-streamed video feed and correlate that against observed radio traffic. Upon detecting the injected signal in camera traffic, it can be deduced that the drone must be observing the target. These systems achieve perfect detection rates with a false positive rate below 0.1 at a distance of 30m. However, they require greater deployment outlay than ours and rely on specific digital video encoding behaviors in the drone [23].

A large number of commercial drone detection systems are available. Systems such as the confusingly-similar Drone Detector[1] and Drone-Detector[2] are comparatively accessible, but still far too specialist for domestic use. Meanwhile, systems from manufacturers such as Thales[3], SAAB Group[4] and Chess Dynamics[5] are military equipment and completely unsuited. Most of the aforementioned approaches are focused on the protection of high-security facilities; permitting the use of expensive, specialist equipment, which is infeasible in a domestic setting. As we specifically target a domestic setting, we need to be able to use ubiquitous COTS hardware. That is why we focus on using the measures of the drone's RF signal that can be obtained from a wide range of inexpensive consumer products.

In our scenario, establishing line-of-sight to the receiver plays an important role. Xiao et al. [39] studied the detection of line-of-sight/non-line-of-sight (LOS/NLOS) conditions. However, they are focused on an office setting with tighter control over the transmitters and most of their metrics rely on such an indoor environment. Furthermore, the transmitters are static in their work, whereas mobility is an intrinsic feature of drones. The detection of moving transmitters has been investigated by Muthukrishnan et al. [22] which we will revisit in Section 5.

Once a drone has been detected acting maliciously, some set of countermeasures may be deployed. A vast array of countermeasures have been proposed; from shotguns, through flying nets [35], live eagles [1], control-signal jamming [15], Wi-Fi de-authentication attacks [26], flight-controller exploitation [19] and GPS spoofing [15, 31], to interference with on-board gyroscopes via acoustic resonance [33]. In a domestic setting, the course of action could be as simple as automatically shutting the blinds. The relative merits of each approach warrant careful consideration, however they are beyond the scope of this work and we do not discuss countermeasures further — focusing solely on the mechanics of detection.

## 2 BACKGROUND

The market for consumer drones is contested by many different manufacturers, but at time of writing it is dominated by only a few companies, namely DJI Innovations (75% market share), Yuneec (5%), 3D Robotics (3%) and Parrot (2%) [27].

The majority of drones provide a live video stream back to the operator; so-called "first-person view" (FPV). In all but the simplest of flights it is a great benefit to the pilot to be able to see from the perspective of the drone itself. The user can commonly connect to the drone with their smartphone, tablet or a dedicated controller to receive the video, and often record the footage in higher resolution as well. Interoperability with existing user devices has played a great role in enhancing the appeal of these consumer UAVs. While some higher-end models use proprietary communication technologies for their video downlink, most drones rely on common 2.4GHz or 5.8 GHz Wi-Fi instead. Some even use Wi-Fi for the control channel of the drone. In these cases, the drone can be controlled completely by an app. Otherwise, a separate, dedicated controller is needed and the Wi-Fi connection is only used for the live-view video. While we focus on Wi-Fi throughout this study, the approach could be applied effectively for any receivable RF technology as it depends solely on measurement of received power.

[1] www.dronedetector.com
[2] www.drone-detector.com
[3] www.thalesgroup.com/en/squire-ground-surveillance-radar
[4] https://saab.com/air/sensor-systems/ground-based-air-defence/giraffe-amb/
[5] http://www.chess-dynamics.com/auds/

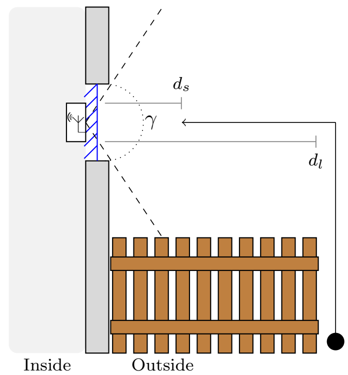



Fig. 2. The attacker launches the drone at launch distance $d_l$ and flies it to surveillance distance $d_s$ to carry out the attack. The detector is installed in the window. The FOV $\gamma$ of the window limits the area that is in LOS of the detector.

Airspace regulations differ between jurisdictions in terms of the requirements placed upon drone operators. Almost all mandate that minimum separation distances are observed between the drone and any surrounding persons or property. In the United Kingdom, the Civil Aviation Authority (CAA) classes drones equipped with cameras as "small unmanned surveillance aircraft" and requires that they [3]:

- fly no closer than 50m from any persons, buildings or vehicles that are not under the control of the operator
- stay below a 400ft flight ceiling
- stay within line-of-sight of the operator

## 3 ADVERSARY MODEL

The attacker uses a radio-controlled drone, with an onboard camera and a live video feed, to invade the privacy of individuals inside a building. Specifically, the attacker uses a drone to look into a window that they would otherwise have a poor view of due to distance and/or height of it. Whether they are motivated by voyeuristic pleasure, a desire to observe private information or even simple curiosity, high-quality camera footage is the means to satisfy their aim. It is also necessary to stream live footage back to the operator throughout, in order for them to pilot the drone accurately[6].

The attacker does not modify their drone to help achieve their goal, although they can fly it in any manner that is required. This is consistent with an opportunist who acts inappropriately with their equipment but lacks the skill, resources or premeditated will to optimize for an attack.

We model the attack as follows; visualized in Figure 2. The attacker does not have access to the premises and is thus positioned some launch distance $d_l$ away from the target window. The launch distance depends on the size of the property and the signal range of the drone controller. They fly the drone from their starting position towards the target window until it can establish line-of-sight (LOS) into the room and is close enough, at surveillance distance $d_s$, to observe a significant portion of the interior at the required quality. Figure 1 compares the drone's viewpoint at

[6]In our experience during data collection, this requirement was surprisingly stringent. Even at small distances it quickly becomes difficult to estimate the relative positions of the drone and obstacles in 3D space from only a single viewpoint. Most relevantly, we found it particularly challenging to know whether a drone had a clear shot of a window without watching its 'first-person view' (FPV) feed. Indeed, not paying due attention to this feed and flying based on the pilot's view alone quickly led to the most serious crash we encountered, wherein the drone contacted the building wall and fell 9 feet to ground, immobilized.

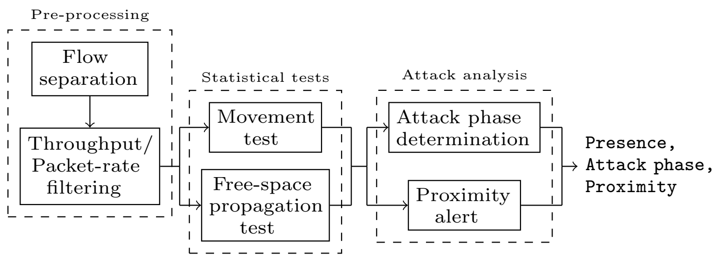



Fig. 3. Flow diagram of the detection algorithm.

different distances. The quality of the footage is determined not only by the distance, but also the camera resolution and the field-of-view (FOV) that the window allows, taken as $\gamma$. The speed of the approach is limited only by the drone's capabilities and so its value $v_{max}$ represents an upper bound on the speed of contemporary drones.

An attack is divided into three distinct phases:

*Approach.* The drone is launched and flies from the start location towards the target window via some flight pattern. The flight pattern may be dictated by obstacles between the attacker and the target, or at the attacker's discretion in an attempt to avoid detection. At some point during the approach, line-of-sight must be established in order to permit surveillance and the drone must get close enough to have a detailed view inside the building.

*Surveillance.* The drone establishes a good surveillance position, it hovers for a period of time and records video of the building interior. The movement of the drone in this phase is kept minimal in order to increase the quality of the recorded footage.

*Escape.* Once the surveillance has been conducted, the drone is piloted away from the window and returns to a collection site.

While the attacker cannot alter the fundamental operation of their drone, they can vary the flight pattern and speed in an attempt to avoid detection. For example, they may try to approach as quickly as possible or as slowly as possible. They may attempt to mimic hobbyist flight or stay out of sight of the window for as long as possible during approach. The attacker is limited only by the bounds of the drone's maneuverability.

## 4 SYSTEM MODEL

A radio receiver is mounted in the building interior, configured to passively listen for all available traffic and record the received signal strength (RSS). This is generally-applicable to any radio technology, but the most straightforward example is a Wi-Fi receiver operating in monitor mode and hopping channels to cover the entire band. We consider that implementation throughout this paper. While an ideal receiver position is at the target window itself, as in Figure 2, the system does not constrain the receiver position; it need not be located in the same room as the target window, nor even within line-of-sight.

With an input stream of traffic and associated RSS values, the system then proceeds in three broad phases, as shown in Figure 3. In the *pre-processing* phase, the received traffic is separated into flows and filtered based on throughput.

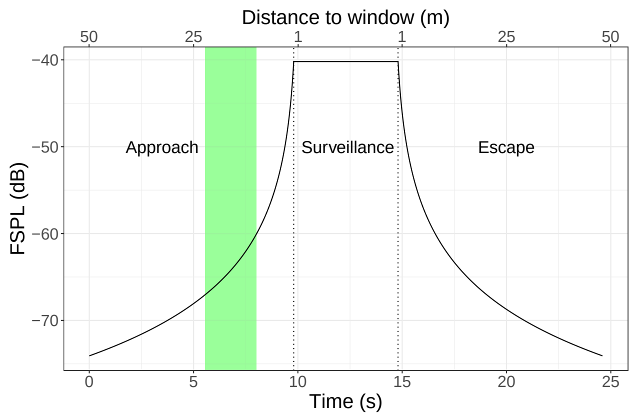



Fig. 4. Simulation of the Free Space Path Loss during a privacy invasion attack with $5s$ surveillance. The speed of the drone is $5m/s$, the surveillance distance $d_s$ is $1m$, the green shaded area shows the time window during the approach when the drone can get detected if a detection distance $d_d$ of 10m is chosen. The detection distance is the distance at which the system is configured to trigger a drone proximity alert.

During the *statistical tests* phase, the system computes live statistics for each flow, derived from the RSS measurements, over a set of rolling time windows of various sizes. It also monitors the baseline statistics of packets from a known, static transmitter (such as the user's Wi-Fi access point). Analysis is performed on the statistical metrics across multiple time windows, along with knowledge of the baseline, to establish whether individual transmitters are moving and operating in free space, consistent with drone behaviour. If the statistical tests pass then the system decides that the moving transmitter is indeed a nearby drone. Subsequently, the *attack analysis* phase determines the attack phase that is in progress and whether the drone has reached a pre-defined detection distance $d_d$ in close proximity.

We expand on each stage below and present specific implementations in Sections 5 and 11.

### 4.1 Pre-processing

The first step towards detection is the separation of different data flows. Given the assumption that the drone is unmodified and that communication makes use of a standard protocol (e.g., the IEEE 802.11 Wi-Fi standard), flows are separated by MAC address. Note, that if the MAC address range of a drone vendor is known, a MAC filter can be a helpful addition to the system. However a MAC filter is not sufficient to detect a privacy invasion attack as a thus detected device could simply be an innocuously operated drone nearby. Instead, our system detects the approach of a drone as well as its proximity to the protected site. Furthermore, as new models of drones are released regularly and because new vendors frequently enter the market, a MAC-based approach would have to be constantly updated to ensure that all drones can be detected. Our system therefore does not rely on a MAC filter based on vendor address ranges, but can be easily extended to incorporate one. The provision of a continuous, live video stream from the drone to the controller, necessitates a consistent, high bandwidth utilization on the communication channel[7]. The detection process excludes flows that do not display this characteristic. Note that this is not a sufficient condition for drone video streams and thus only used as a pre-filtering technique to avoid unnecessary computations. To distinguish drones from other moving or stationary transmitters, the statistical tests described in the following subsection are used.

[7]For example, a Parrot Bebop drone creates 400 packets/s, for a throughput of 500KB/s, when providing 720p video at 30 frames-per-second.

### 4.2 Statistical tests

An airborne drone operates largely in free space, and must establish LOS to the window (and thus the detector) in order to conduct a privacy invasion attack. As such the transmission environment is uncluttered and the received signal strength (RSS) can be expected to be dominated by the direct-path component. We neglect multipath effects due to nearby houses and the ground as the drone has this strong, short-range LOS connection. These free-space propagation conditions yield the RSS behaviour shown in Figure 4; illustrating the path loss changes over time for an attack with a shortest-path approach and constant speed.

The system applies statistical tests to a flow to establish whether it is likely to represent a drone or not. The tests are based upon a comparison of the statistical tests when applied over short and long rolling time windows. Intuitively, the concept is as follows: the message rate of live-streamed video is sufficiently high that a drone will move very little between individual messages, but the drone's movement is the dominant factor in changing the signal strength over a longer period. Short-term tests can therefore be used to determine the propagation environment between the transmitter and the receiver, while long-term tests can establish whether a transmitter is moving.

Combining information from both time periods allows the system to identify transmitters that are both operating in LOS and are moving; which are likely to be drones. Other moving transmitters on the ground or indoors can be expected to have more short-term variation due to changing multipath effects. Whereas immobile transmitters in a static environment are expected to have a stable long-term standard deviation and those in a dynamic environment are expected to experience substantial short-term changes.

### 4.3 Attack analysis

All flows considered to be drones are examined to establish whether or not they are being used to mount a privacy invasion attack. The approach behavior relative to the receiver is determined by monitoring the long-term RSS trend; whether it is increasing, stable or decreasing. A proximity test is then applied to drone flows that appear to be approaching or hovering, to determine whether the drone has reached $d_d$ and is about to begin surveillance.

## 5 STANDARD DEVIATION-BASED METHOD

A suitable implementation of the system model's statistical tests and attack analysis can be derived based on the standard deviation of the RSS values for a given transmitter. We describe that derivation here.

To establish a baseline, we define a noise threshold $\sigma$ that represents the standard deviation of the noise the receiver is subject to. This is determined in an automatic calibration phase by measuring the noise from a known transmitter in a static environment (e.g., Wi-Fi access point).

Given the high packet rate exhibited by a live-streaming drone, in a sufficiently short period the drone does not move enough for the change in RSS due to movement to be distinct from noise. An immobile, LOS transmitter in a static environment exhibits changes in signal strength that are primarily due to measurement noise and cross-traffic interference. Hence, we expect a standard deviation that is close to the general noise level $\sigma$, in such a timeframe. Over a long time period, however, the RSS from a moving drone is mainly affected by that movement. So, whilst the standard deviation alone may not be enough to confirm LOS transmission [39], we can additionally use the long-term changes that are detectable by having a higher standard deviation than for immobile transmitters [22].

In order to apply both tests, the system must first determine appropriate short and long window sizes. To find a window size where the change in signal strength stays below the noise level, we consider the part of the approach with the maximal possible change. This change occurs when the drone travels the final stretch at maximal speed before it arrives at surveillance distance $d_s$[8].

We assume that the drone transmits with packet rate $r$, and $d_s$, $v_{\max}$ and transmission frequency $f$ are given. At packet rate $r$ we receive $N = w \cdot r$ transmissions in a time window of length $w$. As a simplification, we use the free-space path loss (FSPL), or the amount of energy that the signal would lose in a free-space environment, to represent the received signal strength of the drone. Then the free-space path loss of each of these $N$ measurements is given by [14]:

$$x_i = 20\log_{10}\left(d_s + \frac{i}{r} \cdot v_{\max}\right) + 20\log_{10}(f) - 27.55$$

The unbiased sample standard deviation can be computed with:

$$s(N) = \sqrt{\frac{1}{N-1}\sum_{j=1}^{N} x_j - \bar{x}}$$

where $\bar{x} = \frac{1}{N}\sum_{i=1}^{N} x_i$ is the sample mean.

We then have to compute the maximal window size for which the standard deviation is below the noise threshold $\sigma$:

$$w_s = \max\{w | s(w \cdot r) < \sigma\}$$

Then $\sigma$ bounds the standard deviation of the random variable $FSPL$ within window $w_s$.

We consider the signal strength as the sum of random variables $FSPL$ for the free-space path loss and $X_{\mathcal{N}}$ for the noise. Then the variance of the sum of these random variables is given by [21]:

$$Var(FSPL + X_{\mathcal{N}}) = Var(FSPL) + Var(X_{\mathcal{N}}) + 2Cov(FSPL, X_{\mathcal{N}})$$

We know from above that $Var(FSPL) < \sigma^2$ and $Var(X_{\mathcal{N}}) = \sigma^2$. Additionally, we know that $FSPL$ and $X_{\mathcal{N}}$ are uncorrelated, as they are independent random variables. This means that $Cov(FSPL, X_{\mathcal{N}}) = 0$ and it follows that:

$$Var(FSPL + X_{\mathcal{N}}) < \sigma^2 + \sigma^2 + 2 \cdot 0 = 2\sigma^2$$

Consequently, the short-term *free-space propagation* test fails when the standard deviation of the measured samples during $w_s$ surpasses $\sqrt{2}\sigma$.

Analogously, we compute the larger window size to detect movement by choosing an expected minimal velocity $v$ and by looking at the first stretch after launching at distance $d_l$. In this case, the free-space path loss for the $N$ corresponding measurements is:

$$x_i = 20\log_{10}\left(d_l - \frac{i-1}{r} \cdot v\right) + 20\log_{10}(f) - 27.55$$

This time, we are looking for the minimal window size for which the sample deviation is above the threshold:

$$w_l = \min\{w | s(w \cdot r) > \sigma\}$$

Again, using the independence of $FSPL$ and $X_{\mathcal{N}}$ it follows that:

$$Var(FSPL + X_{\mathcal{N}}) = Var(FSPL) + Var(X_{\mathcal{N}}) > 2\sigma^2$$

[8]This is unlikely to happen in reality, as it is too hard to control a UAV at that speed so close to the window. Nevertheless, it helps to determine the largest change that is physically possible and caused by movement alone.

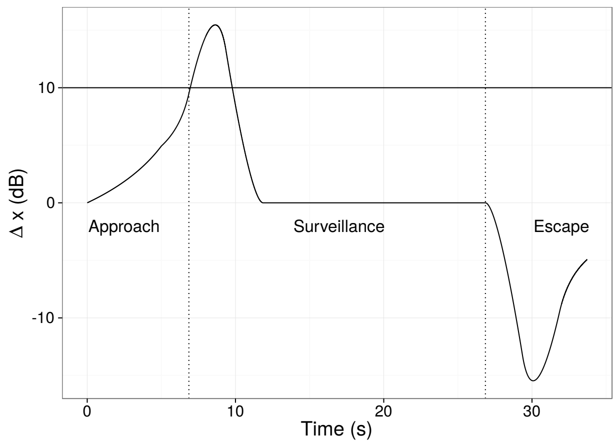



Fig. 5. Attack phase and proximity detection for a simulation with $v = 7$m/s, $d_l = 50$m, Gaussian noise with $\sigma = 2$dB and $\sigma_p = 10$. The black horizontal line is the proximity threshold for $d_d = d_s = 1$m. We used $w_l = 5$s as determined in Figure 9.

As a result, the *movement* test is successful if the measured samples during $w_l$ have a standard deviation that is higher than $\sqrt{2}\sigma$.

### 5.1 Attack analysis

The attack phase can be deduced by taking the difference between the mean of the first and the second half of $w_l$: $\Delta x = \bar{x}_{[1, \lfloor \frac{N}{2} \rfloor]} - \bar{x}_{[\lceil \frac{N}{2} \rceil, N]}$. The sign of $\Delta x$ determines the current attack phase; if the drone is approaching, $\Delta x$ is larger than zero, whereas values lower than zero indicate that it is escaping. If the sign of $\Delta x$ is zero, the drone is not moving. However, this is not sufficient to decide whether the surveillance phase has started. It could just be hovering at a larger distance from the window, without being able to observe the interior. Therefore, we have to take the *proximity* to the window into account. We assume that the approach has been detected previously. Let $w_l$ be the window size in which the detection happened and let $v$ be the corresponding drone speed. Then the drone has arrived at detection distance $d_d$, if $\Delta x$ as defined above is larger than or equal to the following threshold $\sigma_p$:

$$x_i = 20 \log_{10} \left( d_d + \frac{i}{r} \cdot v \right) + 20 \log_{10}(f) - 27.55$$
$$N = w_l \cdot v, \ \Delta x \geq \sigma_p$$

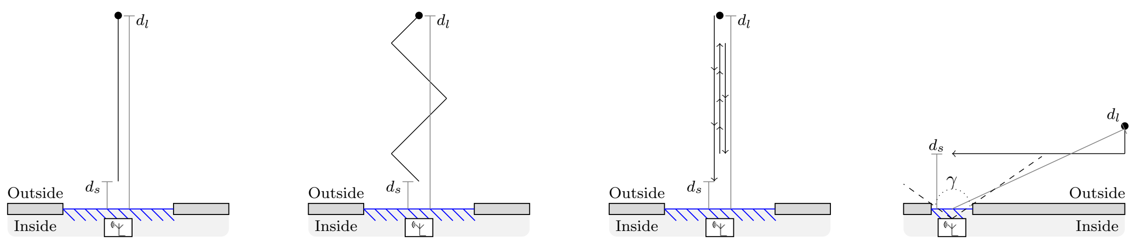
To detect the start of the surveillance phase, $d_d$ can be set to $d_s$. One can see in Figure 5 that the time window has to catch up to detect the surveillance phase. Therefore, it is important that an alarm is already raised as soon as the proximity threshold is breached, rather than waiting for $\Delta x$ to reach zero, to ensure a timely detection. This is especially true, if the approach was very slow as $w_l$ might cover the whole surveillance period otherwise.

### 5.2 Detection range

The detection range $d_r$ is the maximal possible detection distance $d_d$. This is the closest distance to the window that the drone can reach from $d_l$ within $w_l$ seconds. To compute the time until our system can detect an attacker, we take $w_l$ as computed above: $t_d = w_l$. Then the detection range $d_r$ is: $d_r = d_l - t_d \cdot v$. The detection range depends on $d_l$, whereas

Table 1. Example approach flight patterns

| **Approach pattern** | **Description** | **LOS** |
|---|---|---|
| Straight | Drone follows shortest path to window | Constant |
| Zig-zag | Drone follows zig-zag pattern; resulting in variable effective approach speed | Constant |
| Back and forth | Drone approaches, backtracks, then progresses again; resulting in an effective approach speed that is sometimes negative | Constant |
| NLOS | Drone avoids LOS until shortly before surveillance phase | Emerges |




(a) Straight approach (b) Zig-zag approach (c) Back-and-forth approach (d) NLOS approach, with FOV $\gamma$

Fig. 6. Various example approach patterns

the movement speed $v$ has little effect on it, and only shows a slight variation due to the sampling rate. As we do not know the actual speed of the attacking drone in advance, we have to compute a range of different window sizes in parallel to guarantee a timely detection.

### 5.3 Approach patterns

The case studied above corresponds to a straight approach to the window. As noted in Section 3, the attacker can vary their approach in an attempt to avoid detection. We consider a set of example flight patterns; detailed in Table 1 and visualized in Figure 6. Whilst these patterns are not exhaustive, they demonstrate some extremes of the behaviour of an attacker. The first three patterns have in common that the attacker is in LOS of the window for most of the attack; the main difference between these three patterns is the effective speed of the drone or the speed with which the drone closes the distance towards the window as opposed to its actual movement speed. The fourth pattern, the NLOS approach, ensures that the drone is only in LOS of the window for a short period of time. By approaching from the sides, above or below like this, the attacker can give any detection approach, whether human or machine, less opportunity to detect the drone prior to surveillance beginning.

In contrast to the straight approach, flying towards the window using a zig-zag approach results in a lower effective velocity towards the window, which in turn affects the choice of appropriate $w_l$ for the movement test. We simulated the standard deviation for a zig-zag approach with two different sliding windows; one for the actual speed $v$ and another for the effective velocity $v_{eff}$. In Figure 7, one can see that the detection with a window size that is too short for the effective velocity causes a delayed detection. Additionally, the shorter window size captures the zig-zag motion of the approach, whereas these erratic changes get smoothed out by the larger window size of $v_{eff}$. A similar effect is caused

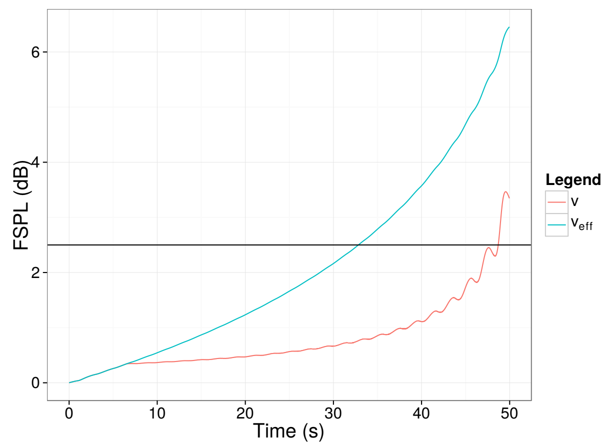



Fig. 7. Simulation of a zig-zag approach with two different choices of movement test window $w_l$. One is optimized for the actual speed, the other one for the effective velocity. The latter allows a faster detection.

by a back-and-forth approach. The choice of $w_s$ for the free-space propagation test, on the other hand, is unaffected by the approach pattern.

If most of the approach is done out of LOS of the receiver, our assumptions about free-space propagation do not hold for the larger part of the approach. Therefore, the part of the approach that can be used for detection with our method gets reduced significantly. This distance depends on the surveillance distance $d_s$ and the FOV $\gamma$ of the window. It is therefore possible that detection might fail under extreme conditions. Depending on the FOV $\gamma$, noise threshold $\sigma$ and surveillance distance $d_s$, the approach in LOS can be too short to result in a notable change of the signal strength. While this is a theoretical possibility, our practical testing suggests it is a remote one. Refer to Figure 8 to see how the detectability of a NLOS approach depends on $\gamma$ and $d_s$.

### 5.4 Detection algorithm

In order to detect drones, we compute the sliding windows for the standard deviation of the signal strength using $w_s$ and increasing values of $w_l$. We proceed like this until we find a value for which the standard deviation is below the threshold within $w_s$ but breaks the threshold for $w_l$.

We then use $w_l$ to determine the attack phase by computing $\Delta x$. For values greater than zero the drone is approaching. If $\Delta x$ additionally exceeds $\sigma_p$, then the drone has reached $d_d$ and an alarm is raised. If $d_d$ is set to $d_s$ then the rolling windows should be reset after a proximity alert to make sure that the surveillance phase is properly detected. The surveillance alarm lasts until the start of the escape phase is detected.

Pseudo-code for the detection algorithm can be found in Algorithm 1, supporting the overall flow diagram in Figure 3. See Figure 9 for the movement and free-space propagation tests of the detection algorithm. This figure shows the rolling standard deviation for a moving transmitter and a receiver that is exposed to Gaussian noise. The attack phase tracking and proximity alert for the same simulation are pictured in Figure 5.

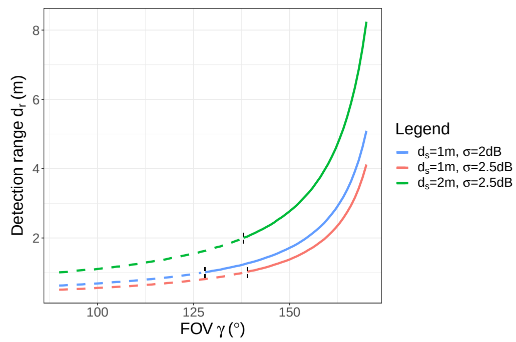



Fig. 8. Detection ranges for the NLOS approach. If the detection range $d_r$ is smaller than surveillance distance $d_s$, the detection will fail.

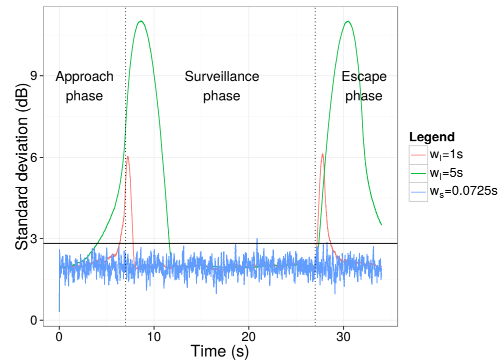



Fig. 9. Detection algorithm for a simulation with $v = 7m/s$, $d_l = 50m$ and Gaussian noise with $\sigma = 2dB$. It computes the standard deviation for the *free-space propagation* test during $w_s$ and several window sizes $w_l$ in parallel for the *movement* test. The black horizontal line is the noise threshold. It is first broken by $w_l = 5s$, making this is preferred window size.

**Algorithm 1:** Standard deviation-based detection algorithm. Here, $s$, *phase* and *proximity* are the tests introduced in Section 5. A full description for the algorithm is given in Section 5.4.

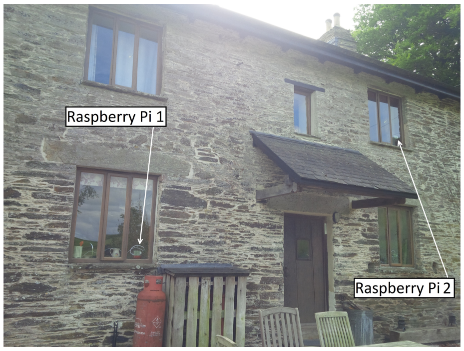
**Data:** Data flow of one transmitter
**Result:** Boolean: Drone alarm, String: Attack phase
**begin**
    Drone alarm, Attack phase $\longleftarrow \emptyset$
    **for** *Increasing* $w_l$ *until* $v_{\min}$ **do**
        **if** $s(w_s) < \sqrt{2}\sigma$ *and* $s(w_l) > \sqrt{2}\sigma)$ **then**
            **if** $phase(w_l) > 0$ **then**
                Attack phase $\longleftarrow$ **Approach**
                **if** $proximity(w_l)$ **then**
                    Drone alarm $\longleftarrow$ **True**
                    **if** $d_d == d_s$ **then**
                        Attack phase $\longleftarrow$ **Surveillance**
                        $reset(w_l)$;
            **else if** $phase(w_l) > 0$ **then**
                Attack phase $\longleftarrow$ **Escape**

## 6 EXPERIMENTAL DESIGN

To demonstrate construction of our detection system using COTS hardware, we made use of a number of Raspberry Pi Model A[9] units as receivers. As this model lacks on-board Wi-Fi hardware, a Wi-Pi USB Wi-Fi adaptor was employed to receive packets. The Raspberry Pis ran the dumpcap utility [10] to capture Wi-Fi traffic, with the Wi-Pi adaptors in monitor mode to allow traffic capture without having to associate to any network.

[9] www.raspberrypi.org
[10] Part of the Wireshark network protocol analyzer (www.wireshark.org)

The real-world experiments took place at a secluded[11] domestic property in Devon, United Kingdom. The property was an old farmhouse with thick, stone walls that were expected to attenuate Wi-Fi signals heavily. The property was surrounded by open land, allowing a variety of approach flight paths.




(a) Outside view

(b) Inside view

Fig. 10. Deployment of the receivers

Two Raspberry Pis were installed in windows at the back side of the house; one in a first-floor bedroom and the other on the ground floor, in a kitchen. Figure 10 shows the deployment locations and one receiver in situ. The Raspberry Pis were connected to a controlling laptop via a Powerline network to avoid introducing cross-traffic on the wireless channel. For calibration, we used the beacons of an accesspoint that was located in the same room as the ground-floor receiver to gather enough calibration packets in a timeframe of thirty seconds. The signal strength measurements of these packets were then used to compute the standard deviation $\sigma$ of the baseline noise, as described in Section 5.

Two very popular drone consumer drone models were used to mount our simulated privacy attacks. The primary candidate was a DJI Phantom 3 Standard, with the experiments being repeated using a Parrot Bebop unit for comparison. Both drone models were operated on the 2.4GHz Wi-Fi band. The two drone models are pictured in Figure 11. Unfortunately, some of the experiments were cut short due to time restrictions and the English weather; especially those from the second series using the Parrot Bebop.

We performed attack runs for each of the approaches described in Section 5; in each case using the first floor window with Raspberry Pi 2 as the privacy-invasion target. The NLOS approach was further split into approaches over the roof and around the house to include both a descent from above and reaching the window from the side. Every approach in the series was performed with three run repetitions. The launch distance was between 55m and 65m for the LOS approaches. For the NLOS approaches the launch distance was much shorter at around 30m as they started from the front side of the house, with the operator moving to keep the drone in view during the approach. No run exceeded a top speed of 7m/s (approx. 25kph/15mph), as it is hard to control the drone at high speeds close to a building. The GPS flight data for each series were recorded by the Phantom. Some example traces are shown in Google Earth[12] plots in Figure 12. One can see that the GPS data became less precise close the window, as the drone no longer had a clear view

[11] To comply with CAA regulations as detailed in Section 2 and avoid disruption to others. The nearest property that was not under our control was over 350m away.
[12] earth.google.com


(a) DJI Phantom 3 Standard

(b) Parrot Bebop

Fig. 11. Drones used in the real-world experiments

Table 2. DJI Phantom 3 experiment series.

| **Approach pattern** | **Distance (m)** Run 1 | 2 | 3 | **Max. Speed (m/s)** Run 1 | 2 | 3 |
|---|---|---|---|---|---|---|
| Straight | 57 | 61 | 61 | 5.94 | 6.24 | 6.35 |
| Zig-zag | 61 | 62 | 62 | 6.36 | 6.66 | 4.72 |
| Back-and-forth | 62 | 61 | 61 | 4.24 | 4.43 | 5.38 |
| NLOS over the roof | 28 | 30 | 30 | 2.78 | 3.92 | 3.62 |
| NLOS around the house | 35 | 31 | 33 | 2.48 | 3.48 | 4.06 |

of the GPS satellites. Specific details of the Phantom experiment series, as extract from the GPS flight data, can be found in Table 2.

## 7 EVALUATION

In this section, we evaluate how well the metrics from Section 5 are able to detect a drone using the data from our experiments. We use only the data from the Phantom for the analysis of the movement test, attack-phase tracking and proximity detection. The results for these are similar for both Phantom and Bebop and, as described in Section 6, we have a more complete set of runs for the Phantom. However due to a technical issue, only the control channel data was captured for the Phantom, so the packet rate is quite low at 40 packets per second. While this is not a problem for the large-window tests, it leaves only about four packets within $w_s$. Hence, we used the data from the Bebop series to examine the free-space propagation test.

In our evaluation, we selected system parameters as given in Table 3. Recall from Section 5 that longer window sizes $w_l$ are necessary to capture slower approaches whereas shorter ones will trigger earlier alerts for faster drones. The system therefore runs with several different window sizes concurrently. The minimal detection range using these parameters was 24.5m.


(a) Straight approach

(b) Zig-zag approach

(c) NLOS approach over the roof

(d) NLOS approach around the house

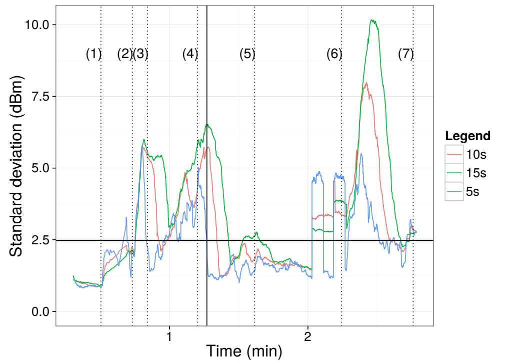
Fig. 12. Different flight patterns during the approach

Table 3. System parameter selections used in Evaluation

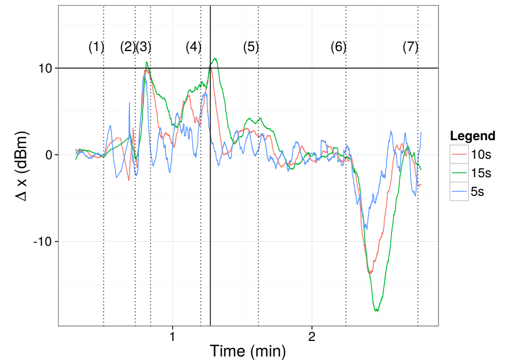
| Parameter | Value | Description |
|---|---|---|
| $d_s$ | 1m | We stayed within one to two meters of the window during the surveillance phase |
| $d_l$ | 50m | All the LOS approaches started further than that |
| $w_s$ | 0.1s | Corresponding to a maximum velocity of 10 m/s |
| $w_l$ | 5s, 10s, 15s, 30s | Corresponding to the following respective minimal velocities: 5.075, 2.535, 1.69, 0.845 m/s |
| $\sigma$ | 1.75 | Derived according to Section 5; hence the noise threshold is $\sqrt{2} \cdot 1.75$ |
| $\sigma_p$ | 10 | The threshold for the proximity alert; derived from $d_s$ |

### 7.1 Straight approach

Examining movement-tracking first, in Figure 13(a) three windows of rolling standard deviation are displayed for the first run of the Straight Phantom set. The drone launched at (1) at a distance of 57m from the window; the first increase is due to the initial ascent, but is not enough to break the threshold. There is a short spike in the 5s window before the actual detection happens which only lasts for a few samples. At (2) the drone started accelerating towards the window. Due to its slow speed at the time, the 10s window is the first to detect it, albeit by a very small margin. The detection happens surprisingly fast after only one and half a meters travelled since takeoff. From the model in Section 5




(a) Standard deviation of the signal strength. The black horizontal line is the movement detection threshold.




(b) Attack phases. The drone is approaching when the values are positive and escaping for negative values. The black horizontal line is the proximity threshold.

Fig. 13. Straight approach with the DJI Phantom 3. The black vertical line shows when the 5s window undercuts the movement threshold. The dotted vertical lines indicate: (1) Takeoff, (5) Surveillance, (6) Escape and (7) Landing. More details can be found in the text.

we had expected the drone to be on target height and approach speed when we received our first packet, however the detection happened much earlier than expected because the takeoff and the varying speed due to acceleration increase the measured standard deviation. In comparison, if we were to start detection during the second acceleration period starting at (3) at a distance of 51m, the drone would be detected 37m from the window. This is closer to the expected value of 25.15m for this distance but still earlier than predicted. This can be attributed to the drone not travelling as close to a constant speed as expected. The velocity decreased at (4), as we had to slow down to pilot the drone safely closer to the house; at distance 6m. The larger window size of 15s compensates for decelerating during the approach and throughout this period the standard deviation stays above the threshold (recall that our system always chooses the lowest window size that is *above the threshold* and uses it for phase tracking and proximity detection). The surveillance period started at (5) where the drone stayed as still as possible in front of the window. After the rolling windows catch up, the standard deviation drops markedly and stays below the threshold almost the entire time until the escape phase starts at (6). However, it does still breach the threshold occasionally. In practice there are still many corrections necessary to keep the drone in place in front of the window. As the drone is so close to the receiver during the surveillance phase, even small movements have a notable effect on the RSS and can breach the detection threshold. Finally, the drone landed at (7) leading to another short spike as it reached the ground.

For attack-phase tracking, we compare the first and second halves of $w_l$, shown in Figure 13(b). Both the 10s and 15s windows stay positive for the entire approach. There is more variation in the 5s window as it captures slowing down better than the longer window sizes. All the windows have values close to zero during surveillance after catching up and are negative during the escape phase, as per our expectations.

A proximity alert is triggered when the phase tracking breaches threshold $\sigma_p$. In Figure 13(b), one can see that with the larger window size of 15s, readings break the threshold early due to the cumulative effect of positive readings over the preceding period. However, at this moment the 15s window is not the smallest for which the standard deviation is above the threshold and therefore another window size is favored over it for phase tracking and the proximity metric —

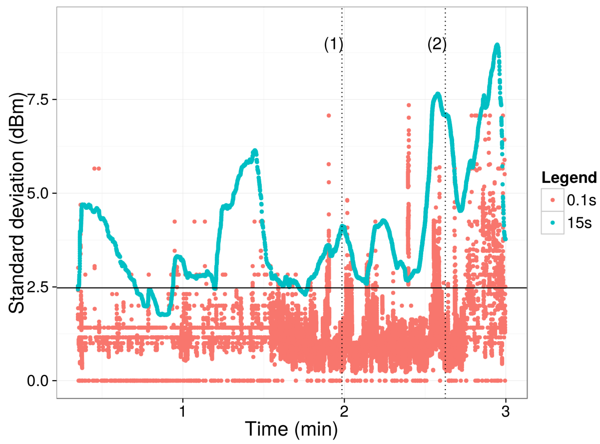



Fig. 14. Standard deviation for a straight approach with the Parrot Bebop. The black horizontal line is the free-space propagation detection threshold which should not be exceeded within $w_s = 0.1s$. The vertical lines indicate: (1) Surveillance and (2) Escape. More details can be found in the text.

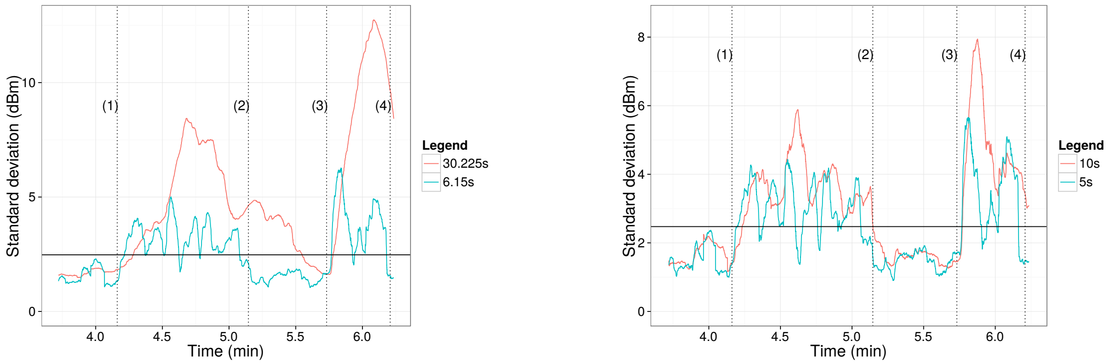
so no spurious proximity warning is produced. The second peak, just after (4), results in a genuine proximity alert for the 10s window. The distance from the window at the time was 4.2m, which is greater than $d_s$. In Figure 13(a), the black vertical line shows the moment when values in the 5s window fall below the detection threshold. Consequently, the 10s window is used to check for proximity and the alarm is triggered.

As mentioned before, we evaluate the free-space propagation test that uses $w_s$ using the Bebop series due to the higher observed packet rate. Figure 14 shows the RSS standard deviation for the first straight Bebop run. The standard deviation of the 0.1s window stays mostly below the threshold throughout the approach; only very few samples violate it initially. When the drone is close to the wall of the building, the standard deviation increases even within $w_s$, hence breaking the threshold more often. At this time, however, we can already be quite sure that it is a drone. The peaks during the surveillance period are caused by movement in front of the window; we had to make more corrections to account for drift with the Bebop, than with the heavier Phantom.

### 7.2 Zig-zag approach

For the zig-zag approach we will first look at window sizes derived from the maximal and effective approach velocity. For the first zig-zag run the maximal approach velocity was $v_{\max} = 5.12$m/s. Using $v_{\max}$ we compute $w_{l,\max} = 6.15$s. On the other hand, the effective velocity for the whole approach was only $v_{eff} = 1.04$m/s and thus $w_{l,eff} = 30.225$s. In Figure 15(a) one can see that the smaller window size detects the drone faster at a distance of 59.5m whereas the larger window size only detects the drone 53m away from the receiver. However, $w_l = 6.15$s captures the erratic changes caused by the zig-zag movement, whereas $w_l = 30.225$s smooths out the flight pattern and shows the whole approach. The decrease in standard deviation in the second part of the approach is caused by a general decrease in approach speed.

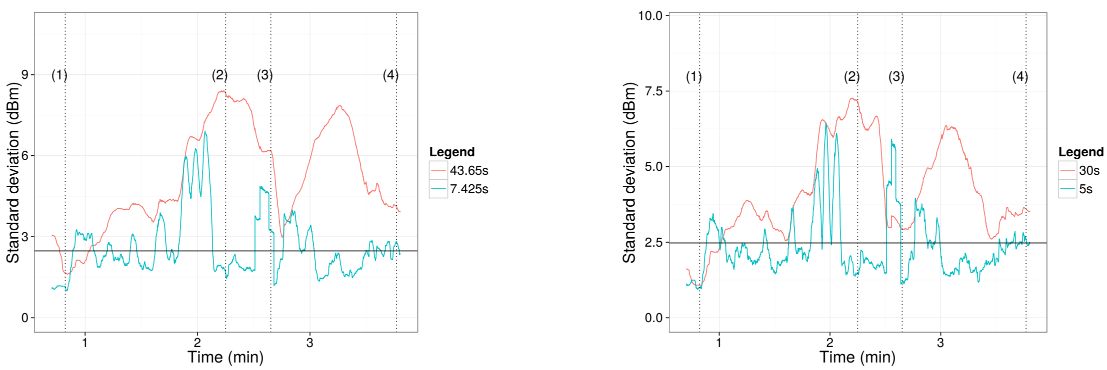
We can see the same effect if we use our system with the parameters from the beginning of the section, cf. Figure 15(b). Lower window sizes than the derived values are possible as $w_l = 10$s already captures the complete approach. This is again likely to be due to an increased standard deviation because of the launch and acceleration periods.




(a) Parameters derived from the model using $v_{\max}$ and $v_{eff}$

(b) System parameters

Fig. 15. Standard deviation for a zig-zag approach with the DJI Phantom 3. The black horizontal line is the movement detection threshold. The vertical lines indicate: (1) Takeoff, (2) Surveillance, (3) Escape and (4) Landing. More details can be found in the text.




(a) Parameters derived from the model using $v_{\max}$ and $v_{eff}$

(b) System parameters

Fig. 16. Standard deviation for a back-and-forth approach with the DJI Phantom 3. The black horizontal line is the movement detection threshold. The vertical lines indicate: (1) Takeoff, (2) Surveillance, (3) Escape and (4) Landing. More details can be found in the text.

### 7.3 Back-and-forth approach

For the back-and-forth approach, we proceed in a similar fashion as for the zig-zag approach. We compute $w_l$ for the maximal speed $v_{\max} = 4.24$s, and the effective velocity $v_{eff} = 0.72$s. Similarly to before, Figure 16(a) shows that the former is affected by the flight pattern whereas the latter captures the complete approach. The shorter window $w_{l,\max} = 7.425$s shows larger peaks followed by smaller peaks. The larger peaks correspond to forward, the lower peaks to backward motion.

In comparison, the higher window size $w_{l,eff} = 43.65$s is able to capture the complete approach as a whole. However, it also covers the entire surveillance period, again highlighting the need for the proximity metric as a stop-and-alarm condition. If we use the system parameters as before, we can see in Figure 16(b) that the 30s window is sufficiently large.

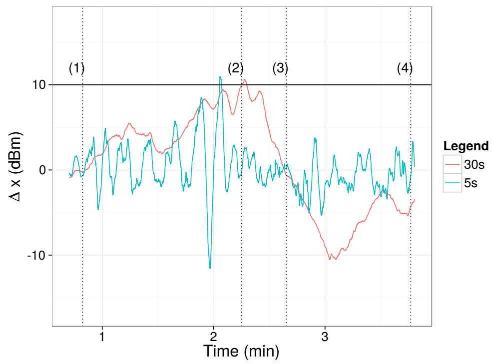



Fig. 17. Attack phases for a back-and-forth approach with the DJI Phantom 3. The black horizontal line is the proximity detection threshold. The drone is approaching when the values are positive and escaping for negative values. The vertical lines indicate: (1) Takeoff, (2) Surveillance, (3) Escape and (4) Landing. More details can be found in the text.

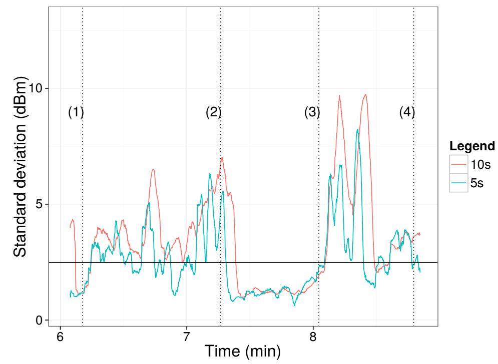



Fig. 18. Standard deviation for a NLOS approach around the house with the DJI Phantom 3. The black horizontal line is the movement detection threshold. The vertical lines indicate: (1) Takeoff, (2) Surveillance, (3) Escape and (4) Landing. More details can be found in the text.

Additionally, we can study the approach phase tracking in Figure 17. We can see that if we only rely on the shorter window, the system oscillates between approach and escape phases, whereas the longer window size makes it possible for us to detect the approach as a larger trend.

### 7.4 NLOS approach over the roof

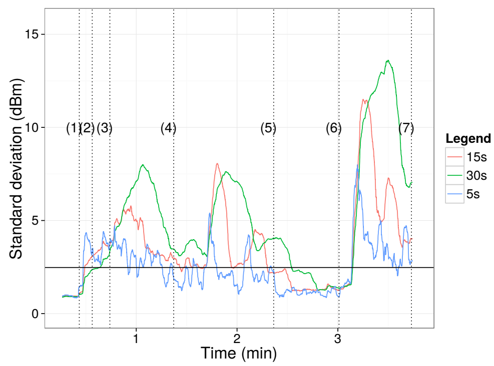
The first NLOS run with an approach over the roof is pictured in Figure 19(a). The drone takes off at launch distance 28m on the other side of the house. Its forward acceleration starts at (2) and it reaches maximal altitude at (3). At (4) the descent begins, on the receiver side of the house. The first big peak for window size 15s happens as the drone is flying over the receiver, above the roof. The next peak occurs as it is descending into LOS, with another one following when it gets closer to the window up to $d_s$. The surveillance period starts at (5), before the drone escapes over the roof again at (6).

We had to fly at slow speeds close to the house for careful maneuvering which explains why higher values are needed to capture the whole approach compared to the straight LOS series.

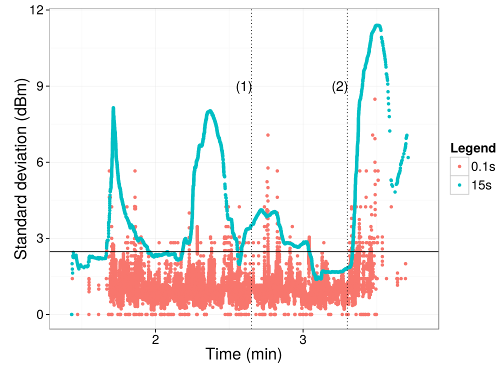
In Figure 19(b), the standard deviation within $w_s$ is displayed. One can see that the standard deviation is low in spite of the missing LOS connection. The roof of the house seems to affect the signal only by attenuation and does not lead to significant interference due to multipath effects.

### 7.5 NLOS approach around the house

A run of the other NLOS approach variant, around the house, is shown in Figure 18. Similarly to the approach over the house, we see peaks as the drone crosses the receiver, gets into LOS and approaches the window. For both NLOS approaches there is a distinct increase in the standard deviation when the drone comes into LOS. Once it is in LOS the behavior is similar to the LOS approaches.




(a) Movement test with DJI Phantom 3. The vertical lines indicate: (1) Takeoff, (5) Surveillance, (6) Escape and (7) Landing. More details can be found in the text.




(b) Free-space propagation test with Parrot Bebop. The vertical lines indicate: (1) Surveillance and (2) Escape. More details can be found in the text.

Fig. 19. Standard deviation for a NLOS approach over the roof. The black horizontal line is the noise threshold.

### 7.6 Detection ranges

To examine detection range we let several rolling windows (5, 10, 15, 30 seconds) run in parallel and chose the first window size to detect the drone as well as the first that stayed above the threshold for the whole approach phase.

Table 4 shows the resulting movement detection ranges. In all of the LOS runs, our system detected the drones far earlier than the minimal detection range of 24.5m, as measured from GPS traces using Google Earth. The detection distances of the second and third back-and-forth runs are lower than the first as they began with a longer and faster forward movement and thus still resembled a straight approach at the time of detection. That the drone flew the approaches with varying velocity actually increased the detection range, as did the more time-consuming flight patterns.

The proximity alert was triggered within 5m of the target window in every experiment run. Unfortunately, due to the roof occluding GPS satellites, the data from the traces is too imprecise to determine the exact distance in each case.

## 8 DISCUSSION

### 8.1 Results

The most notable discovery throughout our experiments was that detection happened earlier than our simulations had predicted. In large part this is simply attributable to our model being very conservative and real-world attacks not demonstrating such extreme behavior. Ever-changing conditions and operator inaccuracy leads real drone flight to have far more speed variance than the constant-speed approaches we had assumed. No two consecutive flights were identical, even with the same intended approach pattern. Close to the target, in the latter stages of an approach and during surveillance, the movement is even more erratic as the drone must navigate a tighter environment and counteract drift from wind conditions and sidewash turbulence in order to establish and maintain a close hover. The reliance purely upon the operator's precise control is intensified as GPS stabilization systems are hampered by a reduced view of the sky when near a building. The detection system parameters must be carefully selected to ensure that the Escape phase is not indicated prematurely.

Table 4. Detection ranges of the DJI Phantom 3 LOS experiment series.

| Approach | Detection range (m) | | | |
|---|---|---|---|---|
| pattern | Run | 1 | 2 | 3 |
| Straight | | 55.5 | 55 | 53.77 |
| Zig-zag | | 60 | 59.5 | 60.13 |
| Back-and-forth | | 60.22 | 51.47 | 48.65 |

Takeoff is the first movement a drone makes, but communication begins before this point. This prior communication is beneficial to our detection as the received messages establish an initial measurement in each window, such that the takeoff itself is enough to bring the standard deviation above the detection threshold in many cases. Such early detection gives ample time for the user to be warned of a privacy-invasion attack and take protective action. It also presents an opportunity to simply inform a user of nearby drone activity that is not, or has not yet developed into, an attempt at privacy invasion.

### 8.2 Observations

Our experimental experiences reinforced the expectation of a reliance by the pilot on the drone's FPV video stream. At close range it is possible (and sometimes preferable) to watch the drone directly when flying, however the benefits of doing this quickly disappear as the drone operates further away and the pilot is less able to reason about its position relative to other objects purely by eye. In this case, and certainly where there is no direct line of sight at all, the attacker depends on the streamed video to pilot successfully. Even in the minimal case, conducting the privacy attack itself, the operator needs immediate feedback to ensure that the drone obtains a good view of the interior of the building — it proved easy to capture detailed footage of a window-frame by mistake. An attacker could, if their hardware permitted, disable the video stream for a period in order to avoid providing a suitable communication stream for the detection system. However, doing so near the target or when attempting surveillance, would likely jeopardize the attack. Indeed, a short surveillance distance is crucial in mounting a successful attack; at large distances a high-resolution camera or telescopic lens is required to capture a detailed image of the building interior and even then the visible area is heavily constrained by the window. Our observations suggest that values of $d_s$ at one or two meters are realistic.

If the drone flies close to the ground for the entirety of the approach, ground reflections are strong multipath components and violate the free-space propagation assumption; making the drone indistinguishable from an attacker moving on the ground themselves. In practice however, we expect that the attacker has to overcome access restrictions and hence must fly higher for at least some part of the approach, making the drone detectable again.

### 8.3 Mobile transmitters

The proposed method can be expected to cope well with static transmitters (e.g., desktop PCs, routers or laptops used in a stationary manner) as variation in their signal strengths is largely due to multipath effects. However, it remains unclear whether it might confuse moving transmitters, especially those outside the house, for drones as they are much more likely to exhibit the characteristics assumed by the model. In the following section, we perform a new set of experiments to determine whether the system is prone to false positives in the presence of (benign) ground-based mobile transmitters.

Table 5. Non-drone approach patterns used in experiment. These patterns were completed using a smartphone. The receivers in LOS of the phone varied between approach patterns.

| Approach | Description | Receivers in LOS |
|---|---|---|
| Front door enter/exit | Enter/exit house through front door | RX 3, 6 |
| Back door enter/exit | Enter/exit house through back door | RX 1, 2, 3, 4, 6 |
| Ground floor | Approach each transmitter on the ground floor | RX 1, 3, 5, 6 |
| First floor | Approach each transmitter on the first floor | RX 2, 4, 7, 8 |
| Stairs | Walk down hallway and up/down the stairs | RX 7 |

## 9 EXPERIMENT WITH MOBILE TRANSMITTERS AND INTERFERING TRAFFIC

We conduct this new set of experiments in the same place and using the same approach patterns in order to ensure comparability. In this new experiment we consider a wider range of receiver placements including less ideal placements without LOS to the approaching drone. Furthermore, we consider both static and mobile benign transmitters in order to investigate likely sources of false positives.

The setup of the experiment, placement of the devices and the routes of the different non-drone approaches can be seen in Figure 20. For the receivers, we used eight Raspberry Pi 4s with a two-antenna Wi-Fi card (Alfa Network AWUS036AC) attached over USB. The receivers were constantly recording Wi-Fi traffic in monitor mode. The drones, static transmitters and the smartphone used as a moving non-drone transmitter were all configured to communicate on the same Wi-Fi channel in the 2.4 GHz band.

Four of the eight receivers (RX 1-4) were placed in windows facing the direction from which the LOS drone approaches were launched. The other receivers were placed in common places around the house: in a bedroom at the back of the house (RX 8), in the downstairs living room (RX 5), at the top floor ceiling similar to a smart light bulb (RX 7), and in the kitchen next to the Wi-Fi router (RX 6).

### 9.1 Non-drone approaches

The non-drone approaches were done using a Huawei P20 Pro smartphone running an iperf2 compatible app creating UDP traffic with a packet size of 750 bytes and a bandwidth of 1.5 Megabytes/second. We considered seven different non-drone approach patterns in which the phone was carried in the hand of one of the authors. These patterns are listed in Table 5 and their paths are outlined in Figure 20. They include entering and exiting the house from the front and the back, starting at a distance of 15m from the house and tracking the drones' approach paths on the way. These patterns were included as they closely resemble a drone approaching the house from the respective side of the house and might therefore be difficult for the system to classify correctly. In addition to these four patterns, we defined three movement patterns inside the house. For each floor we defined a pattern that includes approaches to every receiver and covers every room in the house. Finally, one pattern involved walking down the hallway and going up and down the stairs. We completed at least five runs of each approach pattern.

### 9.2 Static transmitters

For the static transmitters, we used seven Raspberry Pi 4s that were constantly transmitting traffic to a server attached to the router via ethernet. The traffic was generated using iperf2 with the same parameters that were used with the phone for the non-drone approaches (c.f., Section 9.1). By running the static transmitters while the non-drone approaches


(a) Ground floor

(b) First floor

Fig. 20. Approach patterns and equipment placement. The target window of the intruding drone is indicated by a red dashed box. The receiver locations are marked with blue stars, whereas the static transmitter locations are denoted with red dots. Green lines indicate approach patterns.

were performed, we ensured that there was data from a dynamic environment with sufficient line breaks for each transmitter/receiver pair. The static transmitters were placed in locations where one might find high-bandwidth traffic sources (c.f., Figure 20): In the living room, like a voice assistant or smart display, e.g., Google Home or Amazon Echo Show (TX 10), in two of the bedrooms, like a laptop or desktop PC (TX 6, 8), and at the front and back door similar to smart door bells with integrated cameras (TX 1, 2).

### 9.3 Drone approaches

For the drone approaches, we used the same four approach types as before (c.f., Table 1), in each case using the first floor window with RX 2 as the privacy-invasion target. As done in the previous experiment, the NLOS approach was further split into approaches over the roof and around the house to include both a descent from above and reaching the window from the side. Every approach in the series was performed with at least five run repetitions for each of the two used drones (DJI Phantom 3 Standard and Parrot Bebop). The launch distance was between 42 and 50 meters for the LOS approaches. For the NLOS approaches the launch distance was much shorter at around 30 meters as they started from the front side of the house, with the operator moving to keep the drone in view during the approach. No run exceeded a top speed of 7m/s (approx. 25kph/15mph), as it is hard to control the drone at high speeds close to a building.

## 10 EVALUATION OF STANDARD DEVIATION-BASED METHOD

Using the data gathered in this extended experiment, we evaluated the method described in Section 5 by considering static and moving transmitters separately, as well as by taking different receiver locations into account. The evaluation is done for each receiver independently. In this evaluation, each sample of the positive set is a run of a drone approach pattern using the data originating from the approaching drone. By including every approach pattern for both drones this sums up to 51 samples in the positive set. The standard deviation-based method achieves a high true positive rate (TPR) for all of the receivers situated in one of the windows facing the approaching drones. The maximal TPR is 0.98 at

Table 6. Results for the standard deviation-based method from Section 5.

| Receiver | TPR (# FN) | FPR (# FP): static | FPR (# FP): moving |
|---|---|---|---|
| 1 | 0.98 (1) | 0.02 (6) | 0.381 (16) |
| 2 | 0.961 (2) | 0.041 (12) | 0.619 (26) |
| 3 | 0.961 (2) | 0.122 (36) | 0.571 (24) |
| 4 | 0.961 (2) | 0.031 (9) | 0.357 (15) |
| 5 | 0.784 (11) | 0.085 (25) | 0.5 (21) |
| 6 | 0.843 (8) | 0.095 (28) | 0.595 (25) |
| 7 | 0.823 (9) | 0.068 (20) | 0.524 (22) |
| 8 | 0.803 (10) | 0.024 (7) | 0.643 (27) |

receiver RX 1 and none of the window receiver misses more than 2 drone approaches. The results for all the receivers can be found in Table 6.

For the static transmitters, each sample in the negative set is a run of a non-drone approach pattern using the received data of one of the seven static transmitters. In total, this sums up to 294 samples in the negative set. We only consider the time of non-drone approaches for the negative set as this was the only time during the experiment with significant activity within the house. Because the static transmitters are only expected to lead to false positives in a dynamic environment as people move between receivers and transmitters and thus change the signal environment, considering the whole duration of the experiment would skew the results towards a lower false positive rate (FPR). Nevertheless, only few false positives are registered in the case of static transmitters with an FPR of 0.041 or 12 false positives measured at receiver RX 2 in the target window. This indicates that this method is well suited to distinguish drones from static transmitters.

However, this is not the case for other moving transmitters. The negative set for the moving transmitters consists of all the runs of non-drone approach patterns using the data originating from the smartphone. The total size of this set of samples is 42. The performance is far worse in this comparison against moving non-drone transmitters. We measured an FPR of 0.619 at receiver RX 2 in the target window with a total of 26 false positives being registered.

## 11 KURTOSIS-BASED METHOD

To address the issues of the previous method, we designed an improved method based on the kurtosis of short-term signal changes. This is based on the observation that we expect fewer outliers for the signal strength measurements of the drone compared to other transmitters due to the more consistent radio environment and the stable LOS to the receiver in the case of the ideal placement. The kurtosis is the fourth standardized moment and measures how far samples are from the mean. Samples within one standard deviation of the mean contribute almost nothing to the value of the kurtosis, whereas samples that are further away have a higher impact based on distance from the mean. Therefore, a higher kurtosis means that there are more extreme outliers with respect to the standard deviation.

### 11.1 Long-term test

We combine the proximity and approach phase test into a single test. This test triggers when the drone moves from the launch distance to a pre-defined detection distance. For this test we check if the difference of the median of the last few samples within a rolling time window of length $w_l$ and the median of the first few samples breaches a signal threshold $\Delta_p$ that can be derived from the launch distance $d_l$ and the detection distance $d_d$. We use the median to be more robust to singular outliers. The proximity and approach phase test can thus be defined as follows:

$$\text{median}(\text{last } 100ms) - \text{median}(\text{first } 100ms) \geq \Delta_p$$

where

$$\Delta_p = 20 \cdot log_{10}(d_l) - 20 \cdot log_{10}(d_d)$$

In addition to this, we use a similar movement test to before, using the sample standard deviation of the $N$ samples captured within the rolling window. Because of the integrated proximity test, we adapt the method from Section 5 to account for the expected movement of the drone from the launch distance $d_l$ to the chosen detection distance $d_d$.

We know from Section 5 that $Var(FSPL + X_{\mathcal{N}}) = Var(FSPL) + Var(X_{\mathcal{N}})$ and $Var(X_{\mathcal{N}}) = \sigma^2$. Here, $\sigma$ is the noise threshold derived according to Section 5.

$Var(FSPL)$ is given by:

$$Var(FSPL) = Var(\{20 \cdot log_{10}(d_l - i \cdot (d_l - d_d)/N) \mid \forall i \text{ in } 1..N\})$$

Then the movement test succeeds if:

$$s(N) > \sqrt{Var(FSPL) \cdot \sigma^2}$$

For this combined long-term test, several rolling window lengths are used to account for different drone speeds. At each point in time, the window that minimizes the value used in the short-term test described in the following subsection is used for the drone detection.

### 11.2 Short-term test

In the short-term test, we are interested in signal strength changes that are not related to the distance between the drone and the receiver, but are instead due to small-scale fading effects. To minimize distance related changes, we consider the difference between subsequent samples. This is motivated by the observation that the difference in free-space path loss of two consecutive samples is negligible if the sampling rate is sufficiently high.

The sample kurtosis of the $N$ samples captured within the rolling window is as follows:

$$k(N) = \frac{(\frac{1}{N}\sum_{i=1}^{N} \dot{x}_i - \bar{x})^4}{s(N)^4}$$

Here, a sample $\dot{x}$ is defined as the difference of two consecutive samples: $\dot{x}_i = x_i - x_{i-1}$

We consider the signal strength of the drone signal as the sum of random variables $FSPL$ for the free-space path loss and $X_{\mathcal{N}}$ for the noise (c.f., Section 5). Then the sum of these random variables for the difference of consecutive samples is given by:

$$\dot{X} = X_i - X_{i-1} = FSPL_i + X_{\mathcal{N},i} - FSPL_{i-1} - X_{\mathcal{N},i-1} \;= X_{\mathcal{N},i} - X_{\mathcal{N},i-1}$$

As the free-space path loss cancels out, the difference of consecutive samples is the sum of the noise of both samples. If we assume that the noise of each sample is normally distributed, it follows that the difference also follows a normal distribution as the sum of two normal random variables remains normal [22]. Therefore, we expect the kurtosis of the sample differences to have a kurtosis close to three (excess kurtosis close to zero). We expect higher values for the kurtosis of other moving transmitters as they will be subject to other sources of noise and are thus more prone to having outliers.

### 11.3 Message rate

As modern drones are streaming high-quality live video to enable responsive control, a high packet rate can be expected. Therefore, we test if the packet rate is high enough within the tested time period. However, there can be a large amount of packet loss if the drone is still further away. Therefore, this threshold has to be chosen carefully to trade-off detection performance with detection speed.

### 11.4 Algorithm

In order to detect drones, we compute the combined approach/proximity and movement tests described in the previous subsections. This is done using sliding windows with increasing values of $w_l$. When the signal strength increases by more than $\Delta_p$ and if there is noticeable movement as signified by the standard deviation of the signal, we check if the message rate is high enough. Once all these tests are positive, we finally check if the kurtosis of short-term signal differences is lower than the threshold $\Delta_k$. If this is the case, a drone alarm is raised, otherwise the system resumes operation. Pseudo-code for the detection algorithm can be found in Algorithm 2.

**Algorithm 2:** Kurtosis-based detection algorithm. Here, $s$, $k$ and $rate$ are the tests introduced in Section 11. A full description for the algorithm is given in Section 11.4.

```
Data: Data flow of one transmitter
Result: Boolean: Drone alarm
begin
    Drone alarm ⟵ ∅
    for Increasing w_l until v_min do
        if median(last 100ms) − median(first 100ms) ≥ Δ_p and s(w_l) > sqrt(Var(FSPL) · σ^2) then
            if rate(w_l) ≥ Δ_r and k(w_l) ≤ Δ_k then
                Drone alarm ⟵ True
            else
                Drone alarm ⟵ False
```

## 12 EVALUATION OF THE KURTOSIS-BASED METHOD

To evaluate whether the assumption holds that the difference of consecutive drone signal strength samples resembles a normal distribution, we choose an excess kurtosis threshold of 0 for the short-term test. To study the effect of the short-term test and the message rate limitation separately, we first concentrate on the former and disable the message rate check.

We find that we achieve a TPR of 0.902 at a detection distance of 30m with an FPR of 0.095 for the receiver in the target window (RX 2). Both FPR and TPR decrease with decreasing detection distance; the FPR reaches a value of 0 at $d_d = 20$ with a TPR of 0.765, and the lowest TPR is at $d_d = 10$ with a value of 0.569.

If only the long-term test is being used then the TPR is consistently at 1.0 but the FPR is very high ranging from 0.73 to 0.97.

If we look at the histogram of the kurtosis values for a detection distance of 15m in Figure 21, we can see that although the TPR is low if a threshold of 0 is chosen, the kurtosis values for both of the drones are still consistently lower than those of the moving phone.

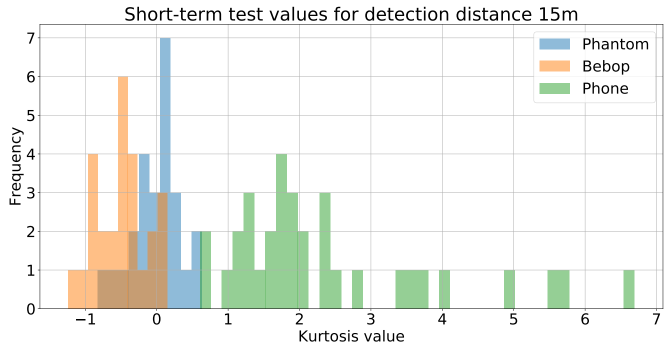
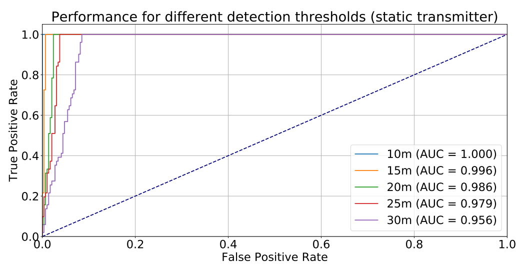
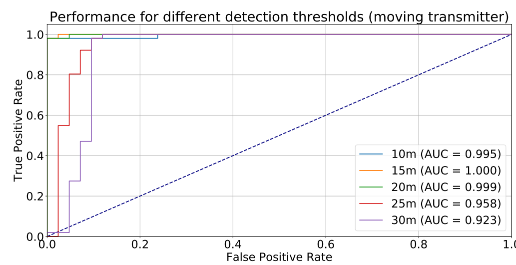



Fig. 21. Histogram of the excess kurtosis values for the two drones and the smartphone with a detection distance of 15m at the receiver in the target window (RX 2). One can see that the kurtosis is consistently lower for both drones compared to the phone.




(a) Static transmitters




(b) Moving transmitters

Fig. 22. ROC curve for various detection distances at the receiver in the target window (RX 2).

## 12.1 Detection performance

If we take this observation into account, we can use higher thresholds on the kurtosis to improve the performance of the detection system. Figure 22 shows a receiver operating curve (ROC) that compares the performance at the receiver in the target window (RX2) for the cases of static and moving transmitters with different threshold values. We observe that very good performance can be achieved for both transmitter types, with closer detection distances exhibiting better performance.

Table 7 shows the area under the curve (AUC) for the receiver in the target window (RX 2) for different approach subsets. Similar to Figure 22, the detection performance is generally higher for detection distances closer to the target window. One can also see that the detection performance increases when only LOS approaches are considered. Furthermore, the system is noticeably better in detecting the Parrot Bebop drone, possibly because of its higher packet rate. Nevertheless, even the performance for the DJI Phantom drone is generally good and no AUC value is below 0.911.

## 12.2 Spatial dependence

The floorplans in Figure 23 show the performance for different receiver locations measured by the AUC when moving non-drone transmitters are considered for a detection distance of 15m. One can see that receivers in the windows facing the courtyard generally perform better as the drone launches from there for the majority of the approaches.

Table 7. Detection performance for the receiver in the target window (RX 2) measured by the AUC for different approach subsets.

| Detection | **All app.** | | **LOS app.** | | **Bebop app.** | | **Phantom app.** | |
|---|---|---|---|---|---|---|---|---|
| distance (m) | Mov. TX | Stat. TX | Mov. TX | Stat. TX | Mov. TX | Stat. TX | Mov. TX | Stat. TX |
| 10 | 0.995 | 1 | 1 | 1 | 1 | 1 | 0.99 | 1 |
| 15 | 1 | 0.996 | 1 | 0.997 | 1 | 0.998 | 0.999 | 0.995 |
| 20 | 0.999 | 0.986 | 1 | 0.986 | 1 | 0.989 | 0.998 | 0.981 |
| 25 | 0.958 | 0.979 | 0.962 | 0.981 | 0.969 | 0.986 | 0.948 | 0.973 |
| 30 | 0.923 | 0.956 | 0.925 | 0.957 | 0.934 | 0.969 | 0.911 | 0.941 |


(a) Ground floor (b) First floor

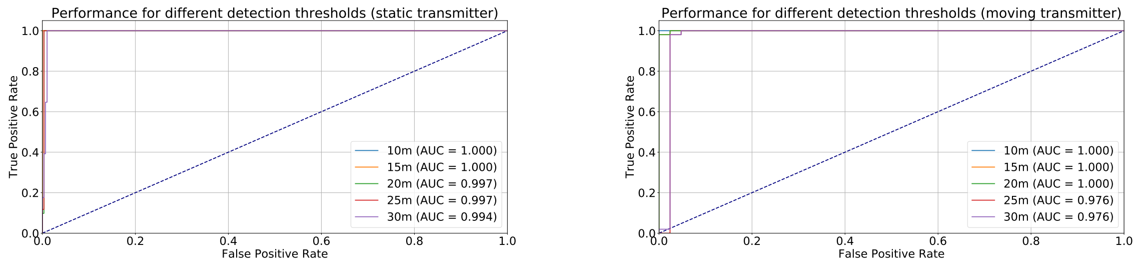
Fig. 23. AUC at different receiver locations for a detection distance of 15m.

Hence, these receivers stay in LOS of the drone for the entire approach for the LOS approaches and come into LOS for the NLOS approaches when the drone reaches the target window. Additionally, receivers on the first floor generally perform better than their ground floor counterparts, presumably because they are subject to weaker ground reflections of the drone Wi-Fi signal.

The receiver in the window on the first floor on the front entrance side (RX 8) exhibits better performance than receivers placed in the middle of the house as it is initially in LOS of the drone for the NLOS approaches.

### 12.3 Multiple receivers

False positives due to static transmitters are mainly caused by humans moving through the direct line between transmitters and receivers. A line break leading to a false positive at one receiver is unlikely to cause a false positive in a different location where the direct line between receiver and transmitter remains uninterrupted. Furthermore, transmitters moving within the house are less likely to keep a stable radio environment between spatially separated




(a) Static transmitters

(b) Moving transmitters

Fig. 24. Performance when more than one receiver is used for detection. Here, the receiver in the target window (RX 2) is used for all tests, whereas the receiver in the adjacent window (RX 4) is only used for the movement test.

receivers. Therefore, we expect that using multiple receivers can improve the detection rate significantly. However, not all of the tests described previously can be applied to more than one receiver at the same time. Tests that include the distance between transmitter and receiver as an implicit factor may fail at one receiver but succeed at the other depending on their spatial separation. Therefore, we use one receiver as the main receiver for the detection, whereas any additional receivers are only used for the long-term movement test.

In Figure 24, we used the receiver in the target window (RX 2) for all tests and the receiver in the adjacent window (RX 4) only for the long-term movement test. For RX 4 we used the movement test from the standard deviation-based method in Section 5 instead of the movement test from the kurtosis-based method as the former test is independent of the distance between receiver and transmitter. There is a considerable performance increase compared to using a single receiver.

### 12.4 Message rate

The detection performance depends upon the received message rate, which in turn depends on the distance from the receiver. Therefore, a message rate threshold has to be carefully chosen with regards to the desired detection distance. If the threshold is too high, no tests will be performed until the potential drone is very close to the transmitter (i.e., late detection). If the threshold is too low, the computation of statistical measures will be poor (e.g., when computing the kurtosis on very few values). We have selected a nominal minimum message rate during our evaluation and we explore the sensitivity of that selection here.

Figure 25 shows the average message rates of the DJI Phantom drone as received by two receivers (receiver RX 2 in the target window on the first floor and receiver RX 5 in the middle of the living room on the ground floor) at different distances. One can see that the average received message rate for RX 2 is below 20 packets per second when the drone is further than 30 meters away from the target window and only starts to increase noticeably as the drone gets closer than 12 meters to the target. For less than ideally situated receivers such as RX 5, the received message rates are much lower and hardly any packets are received when the drone is further than 30 meters away from the target window.

These observations highlight the need for a carefully chosen message rate threshold depending on the receiver location and the desired detection distance.

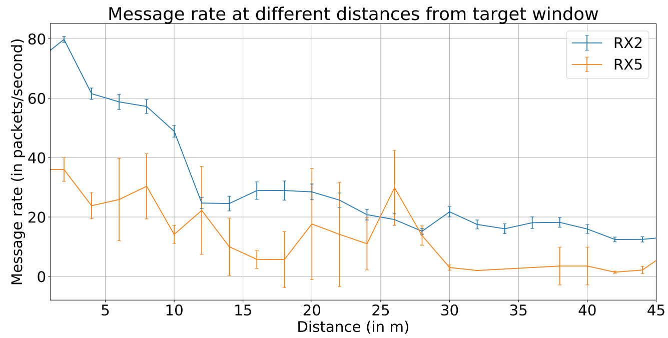
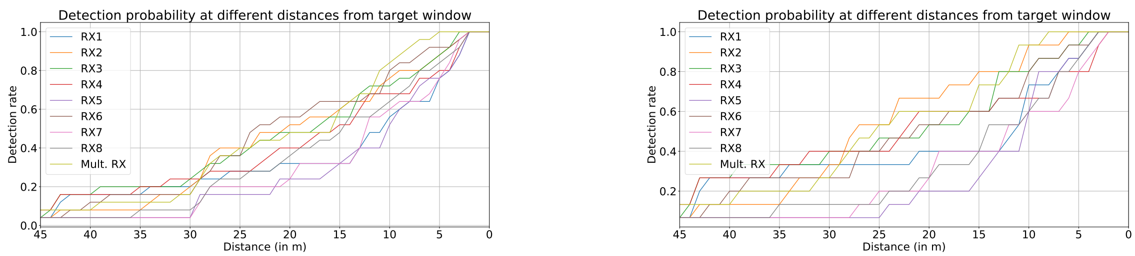



Fig. 25. Message rates at different distances for the receiver in the target window (RX 2) and the worst-performing receiver located in the ground floor living room (RX 5). Error bars represent the 95% confidence intervals.




(a) Including NLOS approaches

(b) LOS approaches only

Fig. 26. Detection probability at different distances with a false alarm rate of zero. Receiver RX 2 is located in the target window and located on the first floor facing the LOS approach site, as does RX 4. Receivers RX 1 and RX 3 are situated below them in the ground floor windows. The other receivers are placed in less ideal positions. The series *Mult. RX* uses receivers RX 2 and RX 4 in the manner described in Section 12.3.

## 12.5 Detection distance

The probability of detection is dependent on how far away from the receiver we want to be able to detect the drone. In Figure 26, the detection probability is shown for different distances when the system is tuned for a false alarm rate of zero. This means that for each meter away from the target window, it shows the ratio of drone approaches successfully identified at or before this point when using an optimal set of system parameters for this detection distance. One can see that the system is able to detect all drone approaches eventually without raising any false alarms. LOS approaches are generally detected earlier than NLOS approaches.

Using the receiver in the target window (RX 2), the system is able to detect all drone approaches at a maximum distance of 2 meters (6 meters if only LOS approaches are considered). This can be further improved when more than one receiver is considered, with a maximum detection distance of 5 meters for all drone approaches (8 meters for LOS) when RX 2 is used in combination with the adjacent first floor window receiver RX 4. For LOS approaches, the majority of privacy invasion attempts are detected from further than 25 meters away from the target window in both cases (single and multi-receiver).

### 12.6 Comparison with other systems

A comparison of our approach to other detection systems is challenging due to the many differences in the considered scenarios, experimental design and reported metrics. Out of all the related works, ADS-ZJU [32], Matthan [25] and the work of Nassi et. al. [23] are the most comparable as they all consider RF-based detection methods, albeit using specialist hardware.

At ranges below 10m, our method outperforms these systems by detecting 100% of drone approaches, with a false alarm probability of 0. As the system was designed to handle privacy-invasion scenarios, it is unsurprising that it is most effective at these ranges — yet notable as no prior work achieves this performance. The performance reduces with increasing distance however and other systems, typically designed for longer-range detection, then perform better. When our system is tuned to match the metrics quoted in other works it displays: recall of 93.3% vs Matthan's 97% recall at 10m with a matched precision of 95.9%, detection probability of 0.73 vs ADS-ZJU's 1.0 at 20m with a matched false alarm rate of 0.028, detection probability of 0.73 vs Nassi et al.'s 1.0 at 30m with a matched false alarm rate below 0.1.

However, our system is purely passive (without an injected visual stimulus as in [23]) and, by contrast, operates with extremely cheap and readily available equipment, such as Raspberry Pis, instead of specialist software-defined radios as all the comparable work does. This means that our system is easier to deploy widely as it can, in many cases, be made available as a software update to devices that are already present in the house without requiring additional hardware.

### 12.7 Limitations

In our system, we consider an opportunistic attacker that uses pre-existing COTS hardware to launch an impromptu privacy invasion attack. A more determined adversary with expert knowledge could attempt to evade the system in various ways.

The attacker could try to disable the live video feed and attempt to control the drone by sight alone. However, we have found that it is very challenging to make the necessary precise maneuvers to stay close to the building if the drone is being controlled from a distance by sight alone. This would likely lead to a high risk of crashing the drone. Furthermore, even just having access to the control channel might make detection possible as the drone is constantly sending telemetry reports back to the controller.

With a pre-programmed flight-path, the attacker could go one step further and disable communication during the attack altogether. But due to the tight manoeuvring that is required for the attack and the imprecision of GPS, especially as the drone can lose the connection to some satellites in the shadow of a building, such an attack is unlikely to succeed in close quarters with the target.

Another option for the adversary could be to modify their drone to use a directed antenna in an attempt to avoid exposing the receivers of the detection system to its radio communication. Aside from the additional technical understanding required to perform this modification, beyond simply the purchase of an off-the-shelf drone described in our threat model, this approach has practical difficulties. Using a directional radio link would make the drone harder to control as the controller would have to be consistently lined up behind the drone irrespective of its rotation, or it would require a complicated and expensive tracking mechanism to compensate. Furthermore, as the drone approaches so close to the system receivers, even the weak sidelobes of a directed transmission may still be enough to pick up the drone's signal effectively.

Finally, we have not tested our system with drones operating in the 5GHz Wi-Fi band. Nevertheless, we expect the detection system to work similarly well for drones using that band. In fact, due to the increased susceptibility of the

5GHz band to NLOS attenuation and the importance within the system of detecting transition between LOS and NLOS environments, it is possible that the detection may prove more effective in this band.

## 13 CONCLUSION

In this work, we developed a method to detect privacy-invasion attacks by drones based on their communication with a controller. Our approach uses analysis of RSS measurements of the drone's transmissions to check for free-space propagation and examine its movement. We developed a further method to monitor if the drone is approaching the detector or moving away from it and another to detect proximity to a receiver. Combined use of these metrics enables us to track the phase of an attack. We conducted real world experiments using two types of commercially-available drones that communicate over Wi-Fi, with various example flight patterns and target windows. For all series, the approaching drone was detected without causing false positives for stationary or mobile non-drone transmitters. We were able to detect a drone approaching in LOS of the detection system at a minimal distance of 6 meters, with the majority of approaches detected at 25 meters. Even for a NLOS approach, detection worked reliably and without triggering false alarms.

In summary, our system is able to detect drones flying nearby and can alert when a drone is in proximity of a window. For even the largest of physical windows, the detection happens at a great enough range that an attacker will have had no chance to conduct detailed surveillance before the alarm is raised. The system is built upon measurements that are available in the vast majority of commodity radio hardware and can operate successfully with receivers in normal, in-home device locations, making it easy to deploy in the real world.

## ACKNOWLEDGMENTS

This work was supported by a grant from Mastercard and the Engineering and Physical Sciences Research Council [grant number EP/M508111/1].

## REFERENCES


[1] Guard From Above. 2016. Intercepting hostile drones. http://guardfromabove.com/ Accessed: 10.07.2016.

[2] Federal Aviation Administration. 2018. FAA Drone Registry Tops One Million. https://www.transportation.gov/briefing-room/faa-drone-registry-tops-one-million Accessed: 08.10.2019.

[3] Civil Aviation Authority. 2016. The Air Navigation Order 2016 And Regulations (CAP393). http://publicapps.caa.co.uk/docs/33/CAP%20393_AUG2016.pdf Accessed: 25.08.2016.

[4] BBC. 2015. Japan radioactive drone: Tokyo police arrest man. https://www.bbc.co.uk/news/world-asia-32465624 Accessed: 08.10.2019.

[5] BBC. 2018. Gang who flew drones carrying drugs into prisons jailed. https://www.bbc.co.uk/news/uk-england-45980560 Accessed: 08.10.2019.

[6] Minas Benyamin and Geoffrey H Goldman. 2014. *Acoustic detection and tracking of a Class I UAS with a small tetrahedral microphone array.* Technical Report. Army Research Lab.

[7] Gabriel C Birch and Bryana L Woo. 2017. Counter Unmanned Aerial Systems Testing: Evaluation of VIS SWIR MWIR and LWIR Passive Imagers. *SANDIA REPORT SAND2017-0921* (2017).

[8] Sanjay K Boddhu, Matt McCartney, Oliver Ceccopieri, and Robert L Williams. 2013. A collaborative smartphone sensing platform for detecting and tracking hostile drones. In *SPIE Defense, Security, and Sensing*. International Society for Optics and Photonics.

[9] Joël Busset, Florian Perrodin, Peter Wellig, Beat Ott, Kurt Heutschi, Torben Rühl, and Thomas Nussbaumer. 2015. Detection and tracking of drones using advanced acoustic cameras. In *SPIE Security+ Defence*. International Society for Optics and Photonics.

[10] Ellen E Case, Anne M Zelnio, and Brian D Rigling. 2008. Low-Cost Acoustic Array for Small UAV Detection and Tracking. In *2008 IEEE National Aerospace and Electronics Conference*. IEEE.

[11] Philip Church, Christopher Grebe, Justin Matheson, and Brett Owens. 2018. Aerial and surface security applications using lidar. In *Laser Radar Technology and Applications XXIII*, Vol. 10636. International Society for Optics and Photonics, 1063604.

[12] T Eshel. 2013. Mobile Radar Optimized to Detect UAVs, Precision Guided Weapons. *Defense Update* (2013).

[13] User 'GWA88' et al. 2018. Gatwick Airport drone incident. https://en.wikipedia.org/wiki/Gatwick_Airport_drone_incident Accessed: 08.10.2019.

[14] Vijay Garg. 2010. *Wireless communications & networking.* Morgan Kaufmann.

[15] Sait Murat Giray. 2013. Anatomy of unmanned aerial vehicle hijacking with signal spoofing. In *Recent Advances in Space Technologies (RAST), 2013 6th International Conference on*. IEEE, 795–800.

[16] Teal Group. 2019. Global drone market estimated to reach $14 billion over next decade: study. https://uk.reuters.com/article/us-usa-security-drones/global-drone-market-estimated-to-reach-14-billion-over-next-decade-study-idUKKCN1UC2MU Accessed: 08.10.2019.

[17] Fraunhofer Institute. 2019. Detection of Small Drones with Millimeter-Wave Radar. https://www.fhr.fraunhofer.de/en/businessunits/security/Detection-of-small-drones-with-millimeter-wave-radar.html Accessed: 08.10.2019.

[18] BI Intelligence. 2017. Drone market shows positive outlook with strong industry growth and trends. https://www.businessinsider.com/drone-industry-analysis-market-trends-growth-forecasts-2017-7 Accessed: 08.10.2019.

[19] Samy Kamkar. 2013. SkyJack. http://samy.pl/skyjack/ Accessed: 10.07.2016.

[20] P. Molchanov, K. Egiazarian, J. Astola, R. I. A. Harmanny, and J. J. M. de Wit. 2013. Classification of small UAVs and birds by micro-Doppler signatures. In *2013 European Radar Conference*. 172–175.

[21] Nitis Mukhopadhyay. 2000. *Probability and statistical inference*. CRC Press.

[22] Kavitha Muthukrishnan, Berend Jan van der Zwaag, and Paul Havinga. 2009. Inferring motion and location using WLAN RSSI. In *Mobile Entity Localization and Tracking in GPS-less Environnments*. Springer, 163–182.

[23] Ben Nassi, Raz Ben-Netanel, Adi Shamir, and Yuval Elovici. 2019. Drones' Cryptanalysis-Smashing Cryptography with a Flicker. In *IEEE Symposium on Security and Privacy (SP), Vol. 00*. 833–850.

[24] B. Nassi, R. Bitton, R. Masuoka, A. Shabtai, and Y. Elovici. 2021. SoK: Security and Privacy in the Age of Commercial Drones. In *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE Computer Society, Los Alamitos, CA, USA, 73–90. https://doi.org/10.1109/SP40001.2021.00005

[25] Phuc Nguyen, Hoang Truong, Mahesh Ravindranathan, Anh Nguyen, Richard Han, and Tam Vu. 2017. Matthan: Drone presence detection by identifying physical signatures in the drone's rf communication. In *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. 211–224.

[26] Matthew Peacock and Michael N Johnstone. 2013. Towards detection and control of civilian unmanned aerial vehicles. (2013).

[27] Skylogic Research. 2018. DJI market share: here's exactly how rapidly it has grown in just a few years. http://thedronegirl.com/2018/09/18/dji-market-share/ Accessed: 08.10.2019.

[28] Artem Rozantsev, Vincent Lepetit, and Pascal Fua. 2015. Flying objects detection from a single moving camera. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. IEEE.

[29] Stephan Sand, Armin Dammann, and Christian Mensing. 2014. *Positioning in Wireless Communications Systems*. John Wiley & Sons.

[30] Linzi Sheldon. 2014. Woman terrified by drone outside her window. KIRO 7. http://www.kiro7.com/news/woman-terrified-drone-outside-her-window/81721261 Accessed: 29.06.2016.

[31] Daniel P Shepard, Jahshan A Bhatti, Todd E Humphreys, and Aaron A Fansler. 2012. Evaluation of smart grid and civilian UAV vulnerability to GPS spoofing attacks. In *Proceedings of the ION GNSS Meeting*, Vol. 3.

[32] X. Shi, C. Yang, W. Xie, C. Liang, Z. Shi, and J. Chen. 2018. Anti-Drone System with Multiple Surveillance Technologies: Architecture, Implementation, and Challenges. *IEEE Communications Magazine* 56, 4 (2018), 68–74. https://doi.org/10.1109/MCOM.2018.1700430

[33] Yunmok Son, Hocheol Shin, Dongkwan Kim, Youngseok Park, Juhwan Noh, Kibum Choi, Jungwoo Choi, and Yongdae Kim. 2015. Rocking drones with intentional sound noise on gyroscopic sensors. In *24th USENIX Security Symposium (USENIX Security 15)*. 881–896.

[34] Der Spiegel. 2013. Merkel Buzzed by Mini-Drone at Campaign Event. https://www.spiegel.de/international/germany/merkel-campaign-event-visited-by-mini-drone-a-922495.html Accessed: 08.10.2019.

[35] Michigan Tech. 2016. Robotic Falconry. http://me.sites.mtu.edu/rastgaar/home/news/ Accessed: 10.07.2016.

[36] Financial Times. 2019. China's DJI targets agriculture as consumer drone sales slow. https://www.ft.com/content/afa5e042-4c50-11e9-bbc9-6917dce3dc62 Accessed: 08.10.2019.

[37] Juan R Vasquez, Kyle M Tarplee, Ellen E Case, Anne M Zelnio, and Brian D Rigling. 2008. Multisensor 3D tracking for counter small unmanned air vehicles (CSUAV). In *SPIE Defense and Security Symposium*. International Society for Optics and Photonics.

[38] WDRB. 2015. Hillview man arrested for shooting down drone; cites right to privacy. http://www.wdrb.com/story/29650818/hillview-man-arrested-for-shooting-down-drone-cites-right-to-privacy Accessed: 29.06.2016.

[39] Zhuoling Xiao, Hongkai Wen, Andrew Markham, Niki Trigoni, Phil Blunsom, and Jeff Frolik. 2013. Identification and mitigation of non-line-of-sight conditions using received signal strength. In *2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 667–674.