Moat: Adaptive Inside/Outside Detection System for Smart Homes

CHIXIANG WANG, Dartmouth College, USA
WEIJIA HE, University of Southampton, UK
TIMOTHY J. PIERSON, Dartmouth College, USA
DAVID KOTZ, Dartmouth College, USA

Smart-home technology is now pervasive, demanding increased attention to the security of the devices and the privacy of the home's residents. To assist residents in making security and privacy decisions – e.g., whether to allow a new device to connect to the network, or whether to be alarmed when an unknown device is discovered – it helps to know whether the device is *inside* the home, or *outside*.

In this paper we present Moat, a system that leverages Wi-Fi sniffers to analyze the physical properties of a device's wireless transmissions to infer whether that device is located inside or outside of a home. Moat can adaptively self-update to accommodate changes in the home indoor environment to ensure robust long-term performance. Notably, Moat does not require prior knowledge of the home's layout or cooperation from target devices, and is easy to install and configure.

We evaluated Moat in four different homes with 21 diverse commercial smart devices and achieved an overall balanced accuracy rate of up to 95.6%. Our novel periodic adaptation technique allowed our approach to maintain high accuracy even after rearranging furniture in the home. Moat is a practical and efficient first step for monitoring and managing devices in a smart home.

CCS Concepts: • **Human-centered computing** → **Ubiquitous and mobile computing**; • **Security and privacy**;

Additional Key Words and Phrases: Smart-home, Wi-Fi, location sensing, inside/outside detection

1 Introduction

The integration of smart devices into private homes brings a range of security and privacy concerns. In particular, the widespread use of sensors like microphones and cameras raises concerns about unauthorized personal data collection [47] and surveillance activities conducted without consent [17]. People may be concerned if unknown smart devices that surreptitiously monitor their activities were installed in their residences, including private homes or rentals. Reports of hidden devices secretly recording or even live-streaming occupants' activities without consent highlight the gravity of these issues [11, 19, 20]. These practices compromise personal privacy and pose security risks, highlighting the need for a robust mechanism to detect and manage such devices. The unexpected presence of an unrecognized device within a home, perhaps following the visit of non-residents, could reveal the installation of a surveillance device. For example, after hosting a gathering or a visit by service personnel, a homeowner might suspect an unknown device has been planted in their home. In such cases, typical network-scanning tools could reveal nearby Wi-Fi devices, but they cannot confirm whether these devices are inside the home (and thus a possible threat) or outside (perhaps in a neighboring apartment). Deciding whether an unknown device is physically inside a home is especially difficult for rentals or apartment residents who share a common Wi-Fi network. Moat is designed to address these threats, determining whether an unknown device is physically within the home, offering enhanced security and privacy protection. Equally concerning is the scenario where an unknown device outside the home, perhaps one impersonating a legitimate device, begins to communicate with a device inside the home. Such interactions may indicate cybersecurity threats attempting to exploit home networks. An inside/outside detection system can provide a proactive defense that enhances other preventative security measures.

Authors' Contact Information: Chixiang Wang, chixiang.wang.gr@dartmouth.edu, Dartmouth College, Hanover, New Hampshire, USA; Weijia He, weijia.he@soton.ac.uk, University of Southampton, Southampton, UK; Timothy J. Pierson, timothy.j.pierson@dartmouth.edu, Dartmouth College, Hanover, New Hampshire, USA; David Kotz, david.f.kotz@dartmouth.edu, Dartmouth College, Hanover, New Hampshire, USA.

Despite the extensive literature on indoor device localization, determining devices' inside/outside status in a practical way is still unsolved. Prior research either requires cooperation from the device itself [6, 25, 26], or needs a floor plan to understand the physical boundary of the home [37]. We detail such differences in Section 1.2. Making the problem more challenging, unlike device detection, which is often only done once, incorporating devices' inside/outside status into a smart home system requires continuous monitoring. When the indoor environment changes, which is bound to happen over time, recalibrating the system would be a burden on the users. A self-adaptive system is thus needed.

1.1 Research Questions

Based on the motivations and challenges outlined above, we propose the following research questions, reflecting the significance of inside/outside detection and the need for sustained robust performance over time:

- **RQ1:** How can a system quickly, accurately, and unobtrusively determine whether a device is located inside or outside a home without prior knowledge of the home's floorplan or cooperation from the device?
- **RQ2:** How can a system maintain robust performance for inside/outside detection in the face of environmental change such as furniture movement?

To answer these research questions, we propose Moat, a system that continuously monitors wireless communications to determine a device's inside/outside status relative to a home. Moat is easy to install, configure, and use. It employs a novel adaptation mechanism to deal with environmental changes that may otherwise decrease classification accuracy in long-term deployments.

Moat leverages a `hub` and several distributed `observers` deployed in the home, as illustrated in Figure 1. The observers passively sniff radio transmissions from devices within their radio range and extract physical characteristics from those signals. The hub then aggregates this data from the observers and uses it to determine whether a transmission was emitted from inside or outside the home. We then extended the basic approach with a novel periodic adaptation method to accommodate indoor environmental changes like furniture rearrangement, to maintain robust performance over a long term. We evaluated Moat in four different homes with diverse commercial smart devices under diverse scenarios, using three homes for evaluation of the basic system and one home for the periodic adaptation method. Our system successfully determined inside vs. outside with an overall balanced accuracy rate of up to 95.6%. Moat is implemented for Wi-Fi, but could be adapted for other protocols such as Bluetooth, Zigbee, or Thread.

1.2 Observations

We highlight several observations about inside/outside determination.

Inside/outside home matters to security and privacy. The location of a device – whether it's inside or outside a home – can significantly influence its security and privacy implications. For instance, the placement of a security camera brings different security and privacy considerations depending on whether it is located inside or outside a home. A system that detects the presence of an unknown device inside the home should trigger an alert to the homeowner or may institute protective measures on its own. Similarly, the detection of an external device impersonating a legitimate one and attempting communication with an internal device also necessitates the home resident's attention. This inside/outside status determination could help prioritize security and privacy responses. Furthermore, if a previously outdoor smart device is moved indoors, or vice versa, a system should recognize such a change and inform the homeowner, instead of blindly continuing with any pre-configured routines. Residents thus need a tool that helps them identify such changes, and enables them to decide whether to integrate, configure, or exclude these devices from their home. Moat can provide such functionality and identify such changes.

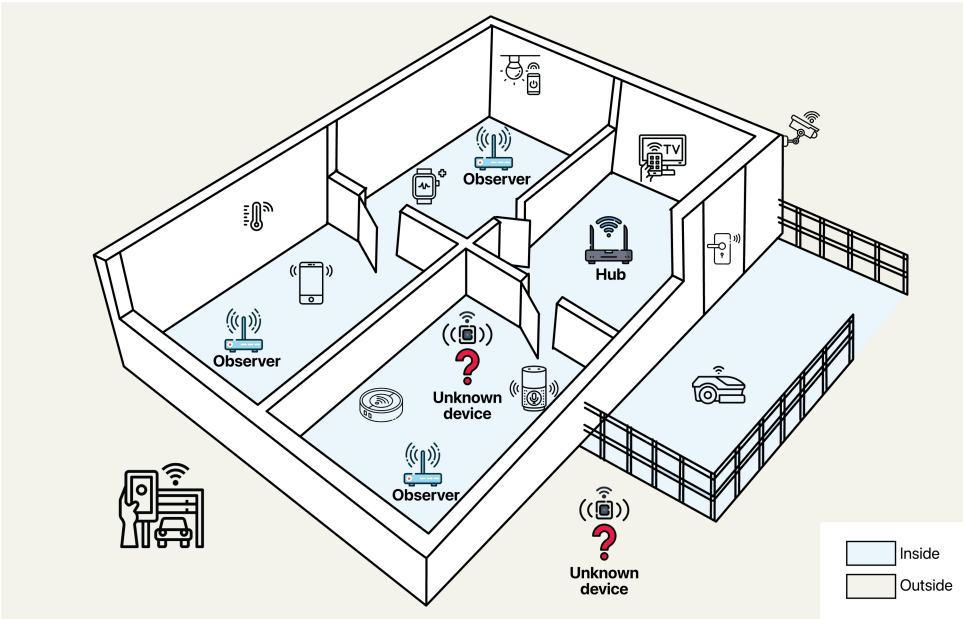

Fig. 1. Application scenario and system overview of the MOAT system on a smart-home network. The system contains a *hub* and multiple *observers* (Wi-Fi sensing devices) that collaboratively sniff Wi-Fi transmissions from nearby smart-home devices and classify them as either *inside* or *outside* the home. We used different colors to show the *inside* and *outside* spaces of a home. The concept of *inside* and *outside* contains both physical and social constructs.

Inside/outside determination is not device localization. Given the extensive literature regarding indoor localization, one might think that the "inside/outside" determination could be resolved by locating the device relative to a map (floorplan) of the residence, and then simply checking whether that location is 'inside' or 'outside' the residence. Not so! MOAT solves a different problem under different assumptions. First, our goal is for the *residence* to determine whether the devices are inside or outside – *continuously, passively, and without requiring any cooperation from the device.* (We cannot expect cooperation from devices that are not part of the home network, or may be adversarial.) Most other localization methods expect a *device* to determine its own location, or depend on the device's cooperation with the system; none of those methods apply here. Second, even for methods where the system determines the location of devices, the system must refer to the location of anchors, or a floor plan for the home, and be able to interpret that plan to decide whether the device is 'inside' the home. However, many home residents do not have, nor do they necessarily want to create, a floor plan for their home. MOAT does not require the user to have or create a floor plan. Third, existing fingerprinting or RSSI and CSI-based localization methods do not consider the influence of indoor environment changes on the localization system. RSSI and CSI are characteristics that are sensitive to the environment, which affects the signal propagation and reflections. We have developed MOAT to self-update to accommodate environmental changes. We discuss more details about why prior device localization techniques fail to meet our goals in Section 2.

Inside/outside distinctions are not always straightforward. The concept of 'inside' and 'outside' of a home encompasses both physical constructs (defined by walls, floors, ceilings), and social constructs (defined by the resident's interpretation of what is 'inside' a home). For example, homeowners likely consider the patio, balcony, and outside garage as private spaces, instead of public – and thus 'inside' their home, despite those areas being

exposed to the outdoor environment. Prior work has not taken such nuances into consideration. Moat allows residents to define their home and what constitutes 'inside' and 'outside' for their specific needs.

Another complicating factor arises when a home with many smart devices is closely surrounded by other homes with their own smart devices. Because radio signals can easily pass through walls, naïve discovery tools will detect devices next door as well as those in one's own home. Mistaking a neighbor's device for a hidden device can result in futile efforts to find the device. Moat quickly determines a transmitter's inside/outside status, alleviating such a search.

1.3 Contributions

This paper presents the design of Moat, with the following **scientific contributions**:

- a base system that can continuously, passively monitor the presence of Wi-Fi devices and accurately determine whether they are 'inside' or 'outside' of a residence – without cooperation from the devices (Section 4);
- an extension that allows it to self-adapt to evolving conditions in the residence (Section 5);
- a further extension to allow that adaptation method to initialize the system with less user effort (Section 5);
- evaluation of the system from the perspective of both basic inside/outside detection (Section 4.3) and long-term adaptation in the face of indoor environment changes (Section 5.2) .

We provide background information in Section 2. We define terms, assumptions, and the security model in Section 3. We describe a basic system design for inside/outside detection and evaluate its performance in three real home environments using 21 devices in Section 4, exploring **RQ1**. We further present a periodic adaptation method to adaptively update the system to accommodate changes in the physical indoor environment and evaluate it in a fourth home in Section 5, addressing **RQ2**. We discuss the limitations and future work in Section 6, and conclude in Section 7.

2 Background & Related Work

Moat leverages physical measurements of Wi-Fi signals – specifically, Radio Signal Strength Indicator (RSSI) and Channel State Information (CSI). By sniffing Wi-Fi traffic from a small number of *observers* distributed throughout the home, and aggregating those Wi-Fi data observations they collect at the *hub*, our system can be quickly trained to continuously distinguish the 'inside' devices from the 'outside' ones. In this section, we provide background information about RSSI and CSI, related work about network sniffing and indoor localization, and related work about smart-home management.

2.1 RSSI and CSI

RSSI refers to the strength of a radio signal measured by a receiver. Radio signals attenuate during propagation, so the RSSI is typically lower the farther the receiver is from the transmitter [10]. Because RSSI suffers from multipath fading and temporal dynamics, it is not a reliable indicator of distance; thus, we also leverage CSI.

CSI is a measurement that describes changes made to a signal in transit between a transmitter and receiver. Changes to the signal occur from attenuation of the signal as it travels and also from constructive or destructive interference resulting from the signal reflecting from obstacles in the environment. These changes are measured in terms of phase and amplitude. Because Wi-Fi uses Orthogonal Frequency Division Multiplexing (OFDM), where data is transmitted on 64 subcarriers[1] simultaneously, the phase and amplitude change of each subcarrier can be measured separately [30]. The CSI measurements across the subcarriers will be different for each observer

[1] A standard 20 MHz Wi-Fi channel uses 64 subcarriers where 52 subcarriers carry data and the others are guard or pilot subcarriers. When multiple channels are bonded together to create a channel with more subcarriers, the same concepts used by our system apply.

in our system because the path from the transmitter to each observer will be different. This characteristic provides our system with a more fine-grained view of the environment than RSSI alone.

2.2 Indoor Device Localization

Indoor localization enables systems to locate people or objects in an indoor environment. We briefly discuss three prevalent methods of device localization to provide context. In Table 1, we compare our inside/outside detection system MOAT against device localization techniques. We choose one example from each of these three device localization methods for comparison. We compare several key factors: whether the system requires a floorplan or spatial geometry information to decide the target devices' inside/outside status, whether the approach needs active cooperation from target devices, the extent of manual effort needed to collect initial training data, compatibility with a wide range of Wi-Fi devices, the system's ability to adapt autonomously to environmental changes that may affect detection accuracy, and whether the technique supports passive continuous monitoring. This comparison highlights the differences in assumptions and applications between determining if a device is inside or outside a home and device localization. In this table, a "target device" refers to any device for which a specific location or inside/outside status is determined by these technologies. We next discuss each of the three localization techniques.

Table 1. Comparison of MOAT (inside/outside detection system), and other state-of-the-art indoor device localization systems.

	MOAT	Fingerprinting (e.g., [40])	Self Localization (e.g., [53])	Geometry-based (e.g., [24])
Purpose	Inside/outside detection		Device localization	
No floorplan or spatial geometry required for inside/outside detection	✔	✖	✖	✖
No cooperation required from target device	✔	✔	✖[1]	✔
No manual data collection	✔[3]	✖	✖	✖
Works with any Wi-Fi device	✔	✔	✖[4]	✔
Adapts to indoor environment changes	✔[5]	✖	✖	✖
Continuous monitoring	✔	✖	✔	✔

[1] The target device localizes itself.
[2] The user needs to hold the searching device.
[3] MOAT needs little manual effort to initialize the system; more details in Section 5.
[4] The device needs to be configured to cooperate with the self-localization system.
[5] MOAT adapts and self-updates with environmental changes.

2.2.1 Fingerprinting-based Localization. Fingerprinting-based approaches enable a system to observe signals from mobile devices and determine their location. RSSI and CSI are often used for this type of localization, where a system attempts to locate a transmitter in three dimensions inside a building [39, 50]; the literature includes multiple excellent surveys [9, 29, 51].

Fingerprinting means that the system uses multiple receivers to measure the signals from a transmitter and to use those measurements to construct a distinctive database of feature vectors extracted from those measurements

based on the transmitter's location. The fingerprint database can be assembled by a moving observer that travels the whole space [2, 39, 40, 42, 44], or by multiple stationary observers that are dispersed in the space [24]. Some prior work uses this method and successfully achieves meter-level indoor localization [8, 45, 49, 54].

Although Moat also uses fingerprinting, we use it for a different purpose: rather than mapping a fingerprint to a set of locations, we map a fingerprint to 'inside' and 'outside'. No prior localization method has yet considered devices' indoor-or-outdoor status; indeed, few such methods even attempt to localize outdoor devices.

A significant oversight of previous fingerprinting-based methods is their reliance on building floor plans to ascertain device positions, which they require to determine the inside/outside status of devices. Our approach overcomes this limitation – Moat can determine a device's inside/outside status without requiring a floor plan.

Furthermore, most Wi-Fi fingerprinting methods aim to allow a mobile device to locate itself, using its own measurements of Wi-Fi signals produced by fixed access points [53]; we flip this notion and develop fingerprints from Wi-Fi measurements made by fixed observers about Wi-Fi signals produced by non-cooperative mobile devices which were not preconfigured to locate itself.

One flaw in prior fingerprinting-based approaches is that they only work in stationary environments or those with very slight changes [27, 39, 46, 52]; that is, the fingerprints depend on the physical characteristics of the environment at the time the fingerprint is measured. Changes in the physical environment, such as the rearrangement of furniture or even the movement of people, reduce the localization accuracy, requiring re-training. Our system gracefully handles these changes without user intervention.

2.2.2 Self Localization. Many methods in current use, and much of the prior literature, focus on enabling devices to determine their own location. For example, a device may use inertial sensors, Bluetooth, or visible light, to determine its indoor location, or observe Wi-Fi and cellular signals to infer its location by triangulation (leveraging a database of base-station locations) or fingerprinting (leveraging a database of signal patterns created by such base stations, correlated with locations) [6, 25, 26, 48, 53]. None of these methods are relevant to our work, because we seek a method for the system (the home's hub) to determine the location (indoor or outdoor) of devices in the area of the home, without any cooperation from those devices.

2.2.3 Geometry-based Localization. Geometry-based localization typically uses multiple antennas to determine signal features like the angle of arrival (AoA) or the time of flight (ToF) [35]. By using geometric algorithms such as triangulation and trilateration, this data can be transformed into location coordinates [24, 31, 33, 52]. For example, Wi-Fi's Fine Time Measurement (FTM) standard derives an estimate of the distance between two cooperating devices by measuring the round trip time of a series of frames exchanged between devices [16]. These estimates, however, require cooperation between devices and are subject to large errors in dense multipath environments like homes [15]. Even if these methods can locate a device relatively precisely, they still require a floorplan to decide whether the device is inside or outside [8]. Our system, on the other hand, does not require such input and does not require cooperation between devices.

2.3 Smart Home Management

With the growing number of smart-home devices in people's households, management of these devices is an increasing challenge. Most current research in smart home management is around enhancing people's awareness of their home status, for example, with dashboards [3, 18, 23]. Such work has focused on providing a detailed analysis of device usage history or energy consumption. Some other proposed smart-home management systems focus more on security and privacy, providing various approaches to enable better transparency and access control [21, 22, 41, 43, 55].

One shortcoming of these approaches is that they all assume a device's location is permanent, ignoring the possibility of an adversary purposefully moving the device. An adversary could bring their victim's outdoor

camera inside, and none of the previously proposed systems would notice. Previous research has shown that the privacy implications of a device can vary depending on its location [4, 32], which means changing the device's location may violate people's privacy expectations. Therefore, our approach can be used in conjunction with previous work. It continuously monitors all devices' indoor/outdoor status, providing an additional layer of awareness of potential security and privacy issues in the home and assisting smart home users in making more informed decisions.

2.4 Smartphone Indoor/Outdoor Detection

Existing research has explored the use of additional software to monitor smartphone-specific sensors – such as light, sound, magnetic field sensors, and cellular signals – to detect whether a smartphone itself is indoor or outdoor [28, 36]. These approaches, however, solve a completely different problem than ours. They aim to make a smartphone perceive whether it is indoor or outdoor, to enhance the smartphone's functionality of context-aware applications. They are explicitly designed around sensors specific to smartphones and do not work for non-cooperative devices that do not run the additional software. This constraint makes these approaches inapplicable to our use case.

In contrast, MOAT aims to passively and continuously determine whether a Wi-Fi device is inside or outside a home without cooperation from the device. MOAT requires only Wi-Fi signals and does not require additional smartphone-specific sensors or cooperation from the target device. Moreover, MOAT maintains robust performance in the face of environmental changes that may affect the light or sound patterns that may decrease the performance of those proposed smartphone indoor/outdoor detection approaches.

3 Terms and Assumptions

In this section, we define terms that are used in later sections. We also define our assumptions about the system, threats, and adversaries to better scope our paper and system.

3.1 Terms & Definitions

Based on the devices' inside or outside state and mobility, we categorize them as follows:

Known device: a Wi-Fi device that is known to home residents, who set up or operate the system. As *known devices* have different mobility, we further categorize them as follows:

- **Known-label:** a device whose inside or outside status is known and does not change, which means we can label it as 'inside' or 'outside' for model training. There are three types of *known-label* devices:
  - ***Anchor:*** a known device that is not expected to move. It may be inside or outside. Examples include fridges, security cameras, garage doors, and weather stations. Anchor devices also include MOAT itself, i.e., the hub and observers, as illustrated in Section 4.1.2.
  - ***Known-inside:*** a known device that may move but is always *inside* home, such as a robot vacuum, indoor toy, or toothbrush.
  - ***Known-outside:*** a known device that may move, but always *outside* home, such as robot lawnmower, automobile.
- **Known-other:** a known device whose inside/outside status changes. It thus cannot be labeled during training or testing. Examples include smartphones, tablets, laptops, and pet wearables.

Unknown device: a Wi-Fi device that is not a 'known device.' The resident may own it (but its MAC address is not recognized or labeled), or not (e.g., owned by neighbors or passers-by). It may be inside or outside home.

3.2 System Assumptions

We assume the smart home and its devices are networked via Wi-Fi. The Moat system components, and the home's smart devices, are connected to the same Wi-Fi network. We assume Moat can query the home's Wi-Fi router to obtain a list of the home's Wi-Fi devices, along with their unique MAC addresses.

As we describe below, the Moat system includes a *hub* device and several *observer* devices; we assume the hub is aware of the observers' MAC addresses and IP addresses, and the observers are aware of the hub's MAC and IP address, enabling them to communicate.

Although not strictly necessary, we anticipate it being helpful for the Moat hub to obtain a list of anchor devices and other known-label devices, and their corresponding MAC addresses. Information about the anchor devices and known-label devices will help enhance the performance of periodic adaptation, which we describe in Section 5. This list may be recognized by the system and confirmed by the resident, or it may be self-reported by the resident.

3.3 Security Model

Although our proposed system has many potential applications, we are especially motivated by its potential to address risks to security and privacy. Thus, we present our threat model and adversary model.

Threats. Smart-home devices pose several potential threats to residents' security and privacy. For example, a visiting contractor may leave a surveillance device inside the home, which, with its sensors, collects sensitive information. Or, a neighbor's device may be compromised and attempt to communicate with the home's devices, attempting to compromise them as well. Our system can discover new devices and alert the home resident before harm occurs. Alternatively, a child or visitor may remove a device from the home without permission, causing the device to be lost or damaged. Our system can note when a device moves from inside to outside.

Adversary. The adversary in this paper may be a visitor with physical access to the interior of the home, a neighbor with the ability to install outside devices within radio range of the inside, or a remote hacker who compromises outside devices to attack the home. We assume in all cases that the adversary is using a device that communicates over Wi-Fi, though not necessarily associated with the home's Wi-Fi router. Regardless of the adversary's goal, we assume it wishes to remain undetected as adversarial; as part of this effort, we assume it aims to defeat our system's ability to determine whether its Wi-Fi signal is 'inside' or 'outside' the home; that is, it desires the system to report 'inside' when the device is actually outside, or vice versa.

We assume the adversary is an unmodified, commercial-off-the-shelf device using standard Wi-Fi protocols. We place out of scope a sophisticated adversary that has the ability to manipulate the device's physical signal (frequency, amplitude, subcarriers) with either physical or firmware modifications. Such attacks are typically complex to implement and less likely to be encountered in real-world scenarios compared to the common adversaries we consider.

4 RQ1 Solution: Basic Inside/Outside Detection

To answer ***RQ1***: "How can a system quickly, accurately, and unobtrusively determine whether a device is located inside or outside a home without prior knowledge of the home's floorplan or cooperation from the device?", we begin by describing a basic approach for inside/outside detection, which we later refer to as the basic Moat, and evaluating its performance in real homes. In Section 5, we present a novel adaptation mechanism to refine this basic approach, which enables it to adapt and maintain accuracy in the face of home environmental changes and reduce manual assistance required for training.

The goal of the basic Moat is to differentiate, in near real-time, which transmitters are located inside and which are located outside the home. During the installation and training process, the system learns the definitions of

the 'inside' and 'outside' areas specific to a home, and establishes a baseline understanding of the corresponding 'inside' and 'outside' signal characteristics. We refer to these transmitters as **target devices**.

4.1 Method

In this section, we describe the system components and the three phases in which the basic Moat operates: installation, training, and monitoring.

4.1.1 System Components. The system comprises two hardware components: a central **hub** for data processing, and several **observers** for passively sniffing Wi-Fi traffic data.

Hub. The hub is a small computing device that serves as the core of the system. The hub can be an independent device, such as a Raspberry Pi, plugged into a wall socket, or be incorporated into an existing physical device, such as the home's Wi-Fi router or a smart-home hub device, as software. The hub receives data transmissions from the *observers* (defined below), pre-processes and aligns the data in time, and trains a classifier for inside/outside detection. The hub can then classify future data from observers and inform the home resident if a target device is inside their home or outside. It also provides a network API so other tools can obtain information about the inside/outside status of devices observed in wireless range, including newly added devices or existing devices that have been moved. The hub is designed to function without extensive storage or computational resources, so it can run on resource-constrained devices such as Raspberry Pi.

Observer. Each observer is a small computing device, about the size of a portable power bank. Similar to common Wi-Fi repeaters, it requires minimal computational resources, but our observers have two interfaces: one to promiscuously sniff frames from one or multiple channels, and the other to communicate a summary of its observations to the hub on a separate Wi-Fi channel to avoid network congestion. Every observer passively sniffs frames, extracts RSSI and CSI from each frame, and sends the measurements to the hub. The frame content is then discarded.

4.1.2 Installation. In the **installation** phase, the home resident physically deploys the hub and observers in the home, and connects them to the home Wi-Fi using either wired or wireless connections. The hub and observers need to be connected for communication and data transmission. The hub can be physically placed at any convenient location within the home. Observers should be positioned as far apart from one another as possible and spread throughout the home, to maximize their signal coverage and encompass different areas of the home. We discuss observer placement in Section 4.3.

4.1.3 Training. In the **training** phase, the system collects training data to build an inside/outside detection classification model.

Data collection. In the basic approach, the home resident performs a single walkthrough of their entire home, both inside and outside, while holding a smartphone running an application that transmits labeled Wi-Fi frames. The observers passively and continuously collect the frames from the smartphone during this stage. The labeled frames then form the ground truth of the Wi-Fi fingerprints from inside and outside the home.

To initiate data collection, the user first taps a button on the app, indicating that they are about to collect data from *inside* the home. The app then starts sending Wi-Fi signals, with the payload containing a label specifying their origin as inside the home. The user then traverses the home while carrying the phone, covering as many inside home locations as possible. The user then taps the button again to terminate the signal emission. The process is then repeated outside their home, with the emitted Wi-Fi frames labeled as *outside* the home.

Data processing. After data collection, the observers extract information from the collected frames – including sequence number, inside/outside label, RSSI, and the amplitude of each subcarrier, which is computed from

CSI. Each observer then creates a feature vector for each Wi-Fi frame received, including the frame's RSSI measurement and 52 CSI subcarrier (excluding pilot and guard subcarriers) amplitude values. To put it more formally, for a Wi-Fi frame i, the observer k creates a feature vector $f_{i,k}$, defined as:

$$f_{i,k} = \{R_i, C_{i,1}, C_{i,2}, \ldots, C_{i,52}\} \tag{1}$$

where R_i is the RSSI measurement for Wi-Fi frame i measured by observer k and $C_{i,j}$ is the CSI amplitude of subcarrier $j = 1, 2, \ldots, 52$ for Wi-Fi frame i measured by observer k.

Each observer k ($k = 1, 2, \cdots, n$, where n is the number of observers) then transmits its own feature vector $f_{i,k}$ of frame i to the hub. On the hub, feature vectors regarding frame i across all observers are concatenated into a single vector F_i:

$$F_i = \{f_{i,1}, f_{i,2}, \ldots f_{i,n}\}. \tag{2}$$

In reality, however, obstacles or distance may prevent a Wi-Fi frame from reaching all observers, causing missing RSSI and CSI values in F_i. Missing RSSI and CSI value means that a part of the observers sniffed the Wi-Fi packets from smart-home devices and extracted RSSI/CSI data from those packets, while the remaining observers did not. In our experiments, the amount of missing RSSI and CSI values accounted for roughly 5% of the total data volume.

We filled the missing RSSI with a dummy value of -100 to indicate the signal is too weak to be received (real RSSI values are usually higher). Similarly, we filled the missing signal amplitude values of each subcarrier from CSI with -1 to differentiate those measurements from the rest (signal amplitude is always non-negative).

Because physical metrics like RSSI and CSI have natural variability even in relatively static environments, the hub smooths the variability by averaging the most recent ω frames as:

$$\overline{F_i} = \frac{1}{\omega} \sum_{k=0}^{\omega-1} F_{i-k}, \text{ where } i > \omega. \tag{3}$$

We use sliding windows with a size of $\omega = 10$ and a step of 5 to average features across consecutive Wi-Fi frames. Each averaged feature vector is labeled with the ground truth 'inside' or 'outside' label. These averaged features are used to train a classifier.

Ensuring a balanced distribution between inside and outside data for training is challenging. To address this issue, we applied resampling techniques to rebalance the imbalanced training data. Specifically, we experimented with two resamplers, namely SMOTE [5] and ADASYN [13]. After evaluation, we opted for SMOTE, the oversampler, as it exhibited superior performance on our dataset.

We fed the balanced training data to a classifier to train an inside/outside model; in Section 4.3, we describe how we selected a specific classification method.

4.1.4 Monitoring. During the **monitoring** phase, the system monitors all Wi-Fi devices and their inside/outside state. The observers passively collect Wi-Fi frames and transmit each frame's physical characteristics to the hub. The hub then uses the trained classification model to infer whether the transmitting device is inside or outside the home. This information can inform security or privacy decisions, or support other applications that benefit from knowing the inside/outside status of devices in and around the home, as described in Section 1. Unlike the training phase, in which the observers report only on frames transmitted by the companion app/device, the observers now monitor and report on frames from ***all*** devices they observe.

4.2 Prototype Implementation

To evaluate our approach, we built a prototype that comprised one hub and eight observers. We intentionally **over-provisioned** the number of observers to evaluate the impact of varying observer quantities and placements.

We used a MacBook Pro as a hub for our prototype. In a real deployment, we envision the hub being embedded in an existing device – such as a Wi-Fi access point, a router, or a smart-home hub like those made by Apple, Amazon, and Google. Although we used a laptop for the hub implementation, its communication and computation requirements can be satisfied by a Raspberry Pi or Arduino, so the hub could also be a small standalone device. Our prototype implementation is transferable to other Unix-compatible servers.

Each of our observers was a Raspberry Pi 4 B with Raspberry OS 5.10.92 and the Nexmon CSI monitoring software tool installed [12, 38]. Each observer sets its built-in Wi-Fi interface to *monitor mode* on the channel used by the home's access point. Monitor mode allows it to capture all frames in radio range, regardless of the frames' source or destination address, even for devices not associated with the home router's SSID. Nexmon allowed us to extract PHY-layer information for each sniffed frame, including RSSI, CSI, sequence number, and source MAC address. The observer transmits this data to the hub periodically, batching data for efficiency. Each observer has a dual-band USB Wi-Fi adapter (Edimax EW-7822ULC or ALFA AWUS036ACM) plugged in. We use its built-in Wi-Fi interface for sniffing and the external one for communication with the hub. The two interfaces use different channels to avoid network congestion (we analyze network congestion in detail in Section 4.4).

During the training phase, for the prototype device that generates frames, we used a Raspberry Pi 4 B running a script that transmits frames at 10Hz.

4.3 Evaluation of the Basic Moat System

To evaluate the performance of the basic Moat system, we tested our prototype at three real homes whose floor plans and structures are shown in Figure 2. Home 1 is a stand-alone, one-floor house, with no close neighbors. Home 2 is a townhouse-like, two-floor apartment sharing one wall with a neighbor and with neighbors in other buildings at relatively close distances. Home 3 is an apartment room in a condominium building, surrounded by other apartments left and right, above and below; one side of the apartment opens to the interior corridor of the building, while the opposite side faces the outdoors.

We placed eight observers in each home, as shown in Figure 2. We do **not** expect that homes of these sizes (less than 150 square meters) will need eight observers, but we purposely over-provisioned the observers so we could experimentally vary the number and placement of observers (Section 4.3.5). While our experiments attempted to spread the observers evenly around the homes, it may not be optimal. Future work will explore automated methods for determining the optimal placement of observers.

To experiment with the system's ability to determine a transmitter's inside/outside state, we selected 21 smart devices of various types to serve as transmitters (target devices). The complete list can be found in Table 2. As illustrated in Figure 2, we deployed the 21 smart devices in the three homes, and each device was positioned in customary locations within the homes based on its category (e.g., the Apple TV was situated in the living room.). The next step was to select a classifier.

4.3.1 Classifier Selection. We evaluated five popular classifiers to determine the best model of inside/outside detection, ideally with minimal computation. The selected classifiers are listed in Table 3. For evaluation, a researcher walked around both the inside and outside of the home with a Raspberry Pi running the training app and collected a training dataset for each home, respectively, as described in Section 4.1.3. The collected datasets are then fed to the five selected classifiers for training with default hyperparameters provided by *scikit-learn* [34].

We conducted cross-validation to evaluate each classifier using test data collected by $n = 8$ observers listening to transmissions from the 21 smart home devices (Table 2) in the three homes. We note that since some devices do not routinely transmit Wi-Fi signals, like the air quality sensor and smart plugs, we used *ping* and *arping* simply as experimental tools to stimulate Wi-Fi echo replies from the target devices and ensure enough traffic during our experiments. In a real user's deployment for long-term monitoring, the *ping* tool would only be used as needed, e.g., for periodic or on-demand inside/outside status checks.

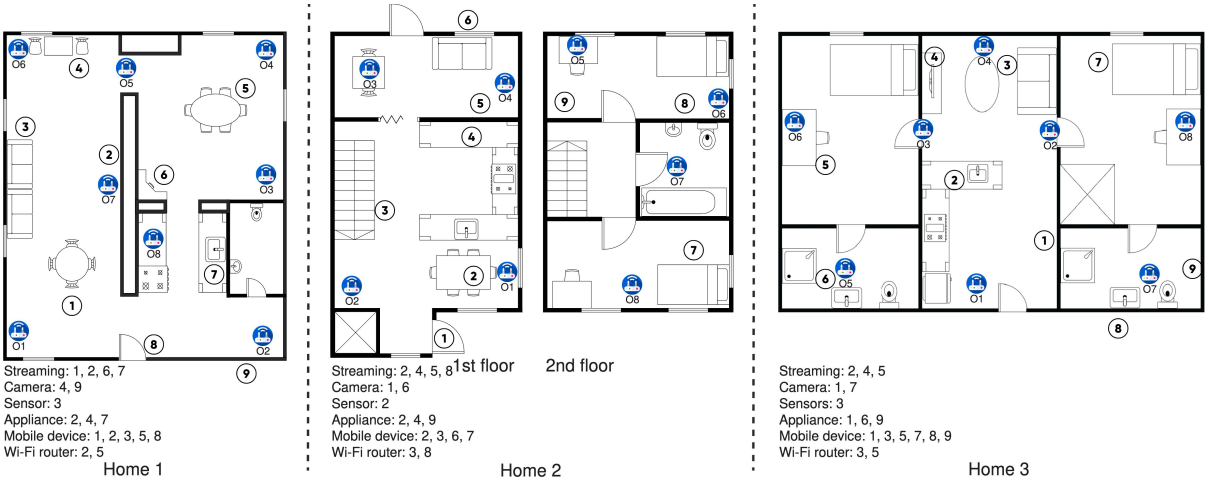

Fig. 2. Floor plans of the three homes, and distributions of the smart devices and observers. The circled numbers in the figure represent the position identifiers, and each position has one or more smart devices placed nearby. The number following the device category name indicates that devices of this particular category are positioned at these designated locations.

Table 2. Smart devices used to evaluate the performance of the inside/outside detection system.

Category	Devices
Streaming	Amazon Echo Dot, Apple HomePod Mini, Google Home Mini, Apple TV
Camera	Blink Security Camera × 2
Sensor	Awair Air Quality Sensor
Appliance	Govee Smart Lamp, Kasa Smart Plug, TP-link Tapo Smart Plug, D-link Smart Plug
Mobile device	iPhone XR, iPhone 13, Nexus 6, Nexus 9, iPad, MacBook Air, MacBook Pro, Raspberry Pi
Wi-Fi router	Netgear router, Alfa Wi-Fi

Table 3 summarizes the results of Home 1. The Random Forest classifier achieved the highest F1 score of 96.3% and balanced accuracy of 95.6%. Across the three homes, the average balanced accuracy for the Random Forest classifier was 94.7%, outperforming other classifiers tested. Because Random Forest performed admirably in all three homes, and it is more computationally efficient than Neural Network (which also performed well), we favored the Random Forest classifier and used it in all the following experiments. Recall that our goal is to deploy this approach in an embedded smart-home hub device, which may have constrained computational resources.

4.3.2 Controlled Grid-based Evaluation. Next, we conducted another experiment to evaluate the system in a controlled manner. In this experiment, we partitioned the inside and outside of a home into a grid of fixed-size square cells and designated the center of each cell as an experimental point. The center of each cell is shown as a box in Figure 3. We then placed a Wi-Fi transmitter (i.e., a Raspberry Pi transmitting Wi-Fi frames 10 times per second) at each of these cells and sent 100 Wi-Fi frames while observers sniffed and recorded the signal characteristics as described above. To ensure fidelity, Wi-Fi frames at each experimental point were labeled with their corresponding point number and whether they were inside or outside for this controlled experiment.

Table 3. Comparison of different classifiers on preliminary experiments conducted in Home 1.

Classifier	Accuracy	Precision	Recall	F1 score
K-Neighbors	0.877	0.934	0.870	0.902
Hist Gradient Boosting	0.889	0.924	0.904	0.914
Support Vector Machine	0.922	0.953	0.910	0.931
Multi-Layer Neural Network	0.934	0.896	0.91	0.922
Random Forest	0.956	0.966	0.96	0.963

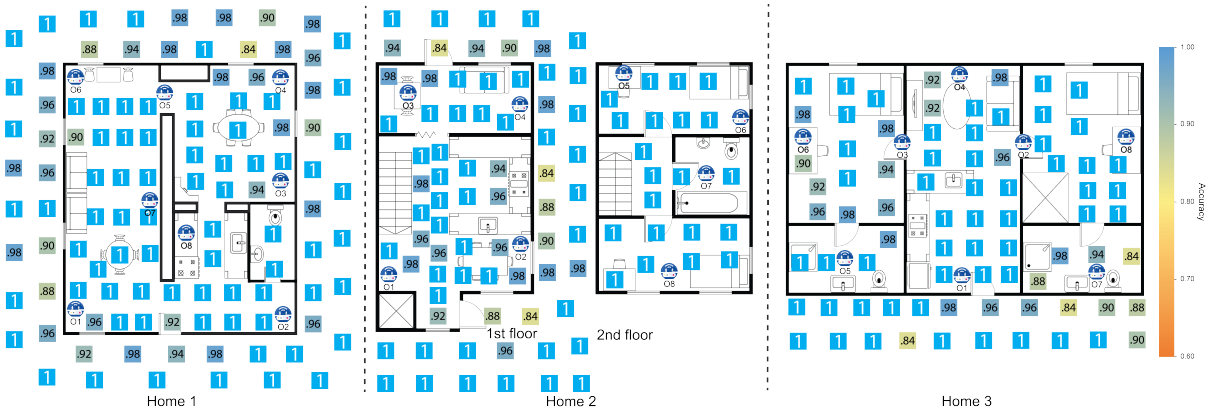

Fig. 3. This figure shows the inside/outside detection system's performance classifying a Raspberry Pi transmitting at different locations inside and outside of two homes, using a Random Forest classifier and eight observers. Boxes indicate transmit locations. The number in the boxes and their color code, as indicated in the color scale at right, represent the classifier's accuracy at rating transmissions from that position. The circular icons represent observer locations.

In this evaluation, we used the trained Random Forest classifier (described in Section 4.3.1), with the grid-based Wi-Fi data from Raspberry Pi as the test dataset. As shown in Figure 3, the trained classifier achieved high accuracy in labeling each experimental point as 'inside' or 'outside' a home at each experiment position. This evaluation was conducted using a single experimental device in controlled settings, resulting in an average accuracy of 98.7%. Locations closer to the center of each home, demonstrably inside the home, were classified with high accuracy.

However, distinguishing locations near the boundary proved to be more challenging. This difficulty arises because our system relies on analyzing differences in Wi-Fi transmission characteristics to determine a device's inside/outside state. Specifically, when two Wi-Fi devices are positioned immediately on opposite sides of a window or door (not a wall) – which are thinner barriers with less signal attenuation than walls, their Wi-Fi characteristics (particularly RSSI) collected by observers are likely to be similar. This similarity makes it more challenging to differentiate their inside/outside status correctly.

Despite this challenge, Figure 3 still shows a robust performance even when a target device is placed near the home boundary. Classification accuracy in these areas only slightly decreases compared to other locations. This consistent observation provides valuable insights into the influence of smart device location on system performance, which is further discussed in Section 4.3.3.

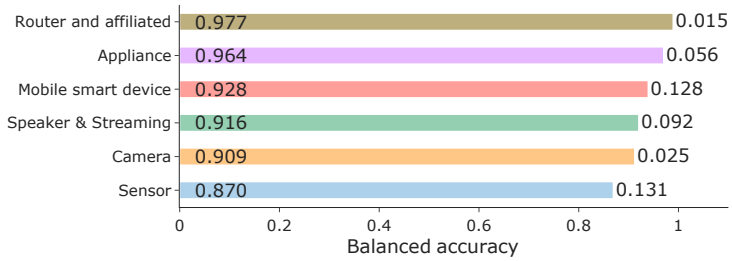

Fig. 4. The performance of the trained RF classifier on different device categories using $n = 8$ observers. The y-axis indicates the names of each device category, and the x-axis value indicates the mean classification accuracy. The number on the left of each bar represents the category's corresponding balanced accuracy. The number outside of each bar represents the standard deviation of that particular category.

The classification performance (of the basic MOAT system) is more related to the distance of target devices to external boundaries such as walls or windows, than the material of the external boundaries. Despite these challenges, our system maintains adequate accuracy even when devices are positioned near windows – areas typically associated with ambiguous classification results. Notably, our experiments conducted in typical residential settings with typical, but different, window sizes affirm that the system's performance remains satisfactory even in such borderline scenarios.

4.3.3 Stationary Device Evaluation. In this evaluation, we explore and evaluate the system performance across different device categories – with commercial smart devices in realistic locations around real homes. We placed the 21 smart devices in locations typical for their types in and around each home (e.g., a TV in the living room), as in Figure 2.

Figure 4 shows the mean balanced classification accuracy of three homes for each smart-home device category, as listed in Table 2. In general, the accuracy of our Random Forest model was over 90% for most categories of smart devices except for *Sensors*. We also noted that the balanced classification accuracy varies between device categories. We speculate that the variations stemmed not only from the device's category, but also from the device's location. On one hand, the device category may influence the accuracy. For example, the air quality sensor transmits scarce and weak wireless signals, which leads to less collected data and, thus, worse classification accuracy. On the other hand, as noted in Section 4.3.2, a device's location influences classification accuracy, due to the nature of the Wi-Fi transmissions and the underlying concept of our system. Some specific devices, such as security cameras and mobile devices, were placed on the door or near the window, which were basically along the home border. For those devices, their location may be the reason for low accuracy.

4.3.4 Moving Device Evaluation. In real homes, some smart devices can move over time. They could move within the home, from inside to outside, or vice versa. We thus evaluated the system's performance on moving devices as they transitioned between outside and inside.

For this evaluation, we selected three commonly used off-the-shelf devices: a security camera, a remote-controlled toy car equipped with a Raspberry Pi to simulate moving devices like robot vacuums, and an iPhone, representing typical mobile smart devices found in homes.

To simulate typical movement patterns of mobile devices, we manually carried the security camera from the inside to the outside, and the iPhone in the opposite direction The toy car roamed indoors, equipped with the Raspberry Pi emitting Wi-Fi frames, to simulate a moving smart device within the home environment.

For off-the-shelf devices that persistently emit Wi-Fi signals, it is challenging to accurately determine the specific interval between two Wi-Fi frames when the device crosses the boundary that separates the inside and

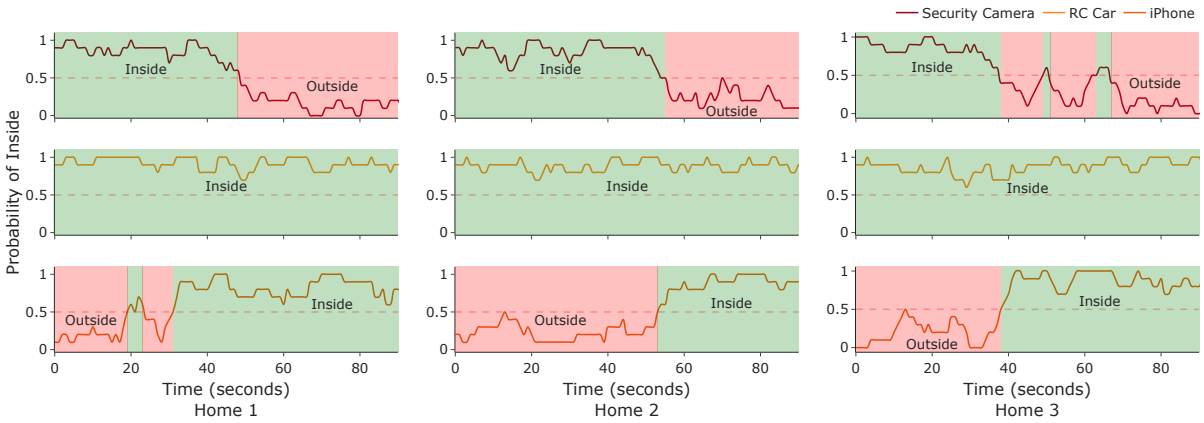

Fig. 5. The classification performance when the system faces moving targets. The x-axis depicts the duration of time over which the device moves, during which it emits Wi-Fi frames at a constant frequency. The y-axis represents the probability that the device is located inside the home, i.e., the classification confidence score for 'inside' state. The green shapes labeled 'inside' show the time period in which the target device is classified as in the 'inside' state, and the red shapes with the label 'outside' show the time period in which it is classified as 'outside.'

outside of a home. Therefore, we cannot label all the Wi-Fi frames, especially those close to the boundary, and form the ground truth accurately. To avoid guessing the labels and introducing bias, we illustrate and evaluate how our system's classification results change over time instead, showcasing the system's response time toward a status change.

Figure 5 shows that for all moving devices tested, our system is able to detect a status change between inside and outside within seconds. It demonstrates that if a user brings a device from outside to inside, or vice versa, our system can recognize the transition and quickly update the device's status accordingly. The probability of inside refers to the confidence score of each classification result for the 10-frame windows. For the remote-controlled car that wandered inside the home, our system never incorrectly recognized it as an outside device.

4.3.5 Number and Location of Observers. As previously mentioned, we installed eight observers in each home to investigate the effect of the number and location of observers. To achieve that, we iterated over all possible combinations of n observers (where n ranged between one and eight), used the data collected from the selected combination for training and testing, and then calculated the system's accuracy for this particular subset. These subsets not only allowed us to investigate how the number of observers affected the results, but also allowed us to investigate different placements of observers and their effects on system performance.

Figure 6 shows the balanced accuracy of different combinations of observers. As expected, the accuracy generally increases as more observers are added. However, after roughly four observers, the improvement becomes less significant with each additional observer. This observation suggests that users do not need to deploy large numbers of observers to obtain accurate classifications. In fact, with just four observers, most combinations of observers can achieve over 92% accuracy in Home 1 and Home 2, and over 83% accuracy in Home 3.

We then analyzed the outliers (shown as whiskers in Figure 6) to understand what kind of observer placement leads to better or worse performance. Figure 7 shows the classification accuracy for Home 2, when using different subsets of 2 observers. Most outliers with lower accuracy are from combinations of observers on the same side of the house and the same floor. This figure suggests that the system's accuracy will benefit from a widely spaced deployment of multiple observers, likely due to the wider diversity of RSSI and CSI values those locations would

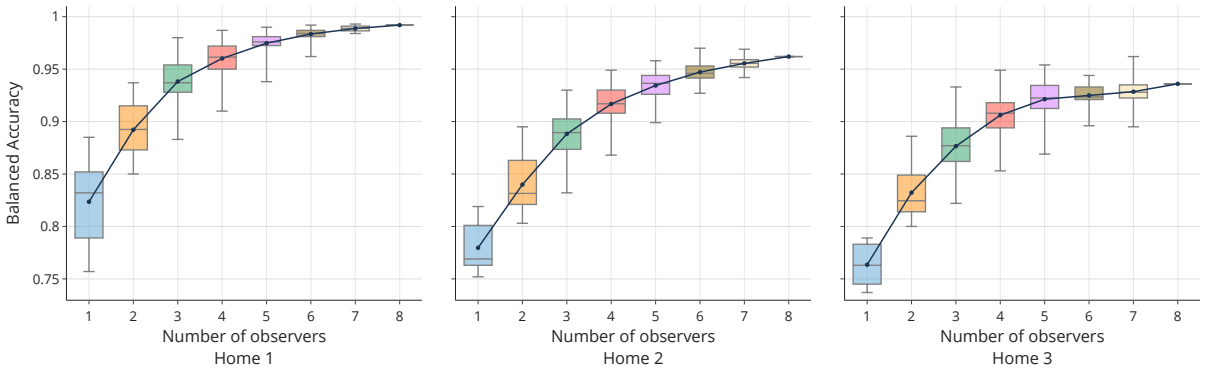

Fig. 6. Boxplot of balanced classification accuracy when using a subset of the eight observers and their corresponding subset datasets as training and testing data in Home 1, 2, and 3. The x-axis represents the number of observers included in each subset, and the y-axis denotes the corresponding balanced classification accuracy. The ends of the box represent the lower and upper quartiles, while the median (second quartile) is marked by a line inside the box. The whiskers show the minimum and maximum values in each subset. A line plot overlaid on the boxplot connects the mean accuracy values for subsets with different numbers of observers, illustrating the overall trend in classification performance.

experience. Through our evaluation, we also believe that the user does not need to follow strict rules to arrange the position of the observers, but only needs to follow a general principle – to make the distribution of these observers as decentralized as possible – to achieve the effect. A full exploration of observer location (and potential methods to automate and optimize location selection) is reserved for future work.

4.3.6 Limited Outside Training Data. In real-world settings, complete access to a home's outside environment is often limited. For example, individual houses may have one or two sides that are inaccessible due to obstructions or occupancy. Similarly, apartments typically have access to only one outside side (a hallway), with the remaining space belonging to other residents or inaccessible outdoor areas far above ground. These limitations prevent a complete outside initial training data collection.

To evaluate system performance under such constraints, we assessed its accuracy with restricted outdoor training data. We used Home 1's data for this evaluation, as Home 1 was the only home where we could collect complete four-sided initial training data. This dataset allowed us to evaluate the system using various subsets of the complete four-sided data, simulating limited-access scenarios.

We trained the model on combinations of subsets of n sides of outside training data (where n ranged from 1 and 4), and subsets with m observers (where m ranged from 1 and 8). The model was then tested with smart-home devices that are stationary or mobile but located strictly inside or outside the home.

Figure 8 shows the balanced accuracy of different combinations of limited outside training data and observers. The results demonstrate a strong positive correlation between the diversity of outside training data and the average classification accuracy. However, in cases where outside training data is limited, the system can achieve an average balanced accuracy of up to 83.7% with more observers, even when trained on *only one* side of outside training data.

4.4 Data Channel Network Congestion

After receiving a Wi-Fi frame transmitted by any target device, each observer extracts CSI/RSSI data from the frame and sends those measurements to the hub through a *separate* Wi-Fi channel; for clarity, we refer here to that

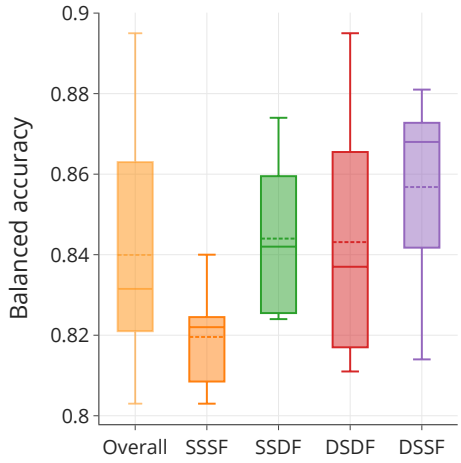

Fig. 7. The distribution of balanced accuracies is achieved by using different combinations of two observers ($n = 2$) within Home 2. The leftmost "Overall" column provides a focused view extracted from the result in Figure 6 of Home 2, where the number of observers is 2. The subsequent four columns of box plots show system performance under different combinations of observers' locations. Each box plot represents a specific combination of observers, categorized by their relative positioning within the home. The mean value is marked by a dashed line inside the box, while the median (second quartile) is marked by a solid line inside the box. The specific meaning of each category is: Same side, same floor (SSSF): Both observers are situated on the same side of the house and the same floor. Different side, same floor (DSSF): Observers are located on opposite sides of the house but on the same floor. Different side, different floor (DSDF): Observers reside on opposite sides of the house and different floors. Same side, different floor (SSDF): Observers are positioned on the same side of the house but on different floors.

channel as the MOAT channel. Because there may be multiple observers, the MOAT channel could theoretically be overwhelmed if target devices transmit rapidly. In the extreme case, target devices transmit small frames that completely fill the Wi-Fi communication channel. With 8 observers, 8 CSI/RSSI measurements would be sent to the hub for each device-transmitted frame. Without a mitigation strategy, the MOAT channel would have to be about 8 times larger than the target-device channel to accommodate this network traffic.

MOAT mitigates this data channel congestion problem by aggregating multiple CSI/RSSI measurements into a single frame to send to the hub. This aggregation approach dramatically reduces Wi-Fi overhead. When a target device transmits a Wi-Fi frame, in addition to the data payload, each frame includes a preamble, header, and checksum. Moreover, the device waits for an acknowledgment from the router before sending the next frame. This non-data overhead is significant. MOAT aggregates 10 CSI/RSSI measurements and their corresponding MAC address into a single frame. In this way, MOAT only sends one preamble, header, and checksum, and only waits for one acknowledgment, eliminating the overhead incurred by 9 target device transmissions.

We provide a detailed analysis of the communication congestion saved by reducing this overhead in Appendix A. We estimate that MOAT can handle 8 observers using a single 20 MHz Wi-Fi data channel, even in the theoretical worst-case scenario where a target device constantly transmits at maximum speed. While MOAT can handle 8 observers in the theoretical worst case, in our experiments we see that 4 observers is sufficient to handle most homes or apartments. We also found real IoT devices transmit at a significantly lower rate than the theoretical maximum.

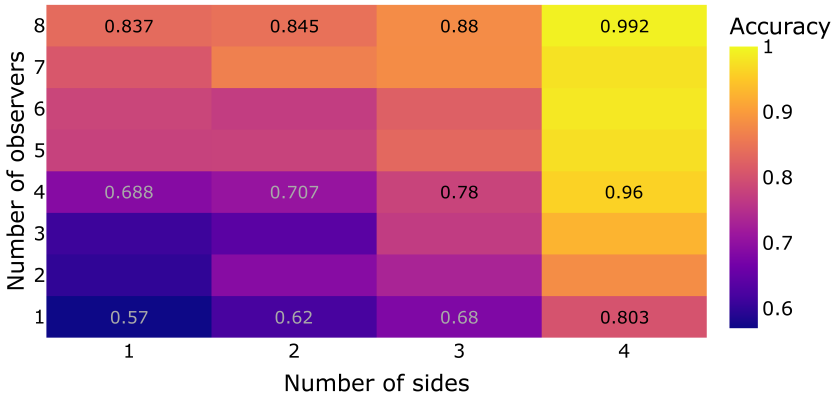

Fig. 8. System performance with different limited amounts outside training data in Home 1. Each heat unit in this figure shows the best value of the accuracies with different combinations of subsets of outside data and subsets of observers. The numbers on the units are the accuracies for each of their corresponding setting. The x-axis means the number of sides in each subset that outside training data were selected and used for model training. The y-axis represents the number of observers in each subset. The heat denotes the average accuracy with different combinations.

4.5 Discussion of RQ1

The evaluation above shows that the basic Moat system can effectively discern devices located inside and outside of a home, addressing ***RQ1***. This section discusses additional aspects of the basic Moat's performance and application.

4.5.1 Limitation of the Basic Moat System and the Need for an Adaptive Approach. While the basic Moat has shown robust performance in real homes, our initial evaluations were conducted with data collected on a single day. Collecting data in one day does not account for potential variations in the indoor environment and their impact on system performance. This limited timeframe does not capture variations in the indoor environment that might affect performance over time, such as the rearrangement of furniture – which alters the multipath propagation patterns learned during the model's training phase. In the next section, subsequent tests in environments with altered layouts demonstrated a decline in the system's accuracy. To address this concern, we introduce a novel periodic adaptation method in the following section, designed to maintain efficacy despite environmental changes.

4.5.2 Manual Effort in Initialization. With the basic Moat system, we can effectively achieve the basic goal of differentiating inside and outside devices. This approach, however, requires home residents to hand-hold a transmission device and walk around their home, both inside and outside, to manually collect training data that allows the system to learn what is 'inside' and what is 'outside' their home. The manual data-collection process is also a major limitation for many other fingerprinting-based techniques [40, for example]. The basic Moat system does not require much effort from the home resident to train and initialize, and only requires them to walk indoors and outdoors with no requirements for routing or duration. The process may only take a few minutes for an apartment or house of average size. Even so, we would like to reduce this burden on the user. In the next section, we propose a new approach that not only automatically updates the system as mentioned above, but also mitigates or even eliminates the process of manual data collection. This extension is another contribution of this paper.

4.5.3 Hardware Considerations and Cost Implications. Moat requires the installation of several observers; this additional hardware imposes a small financial and practical burden on users. The observers are simple devices,

however (our prototype used a Raspberry Pi for each observer), and should therefore be small and inexpensive. The Moat hub could be implemented as software and integrated into the router or even a cloud-based server for processing; it need not require any additional hardware.

5 RQ2 Solution: Periodic Adaptation

The home's environment can change (e.g., due to furniture rearrangement), rendering the previous classification model less effective or unusable. We thus raise ***RQ2***: "How can a system maintain robust performance for inside/outside detection in the face of environmental change such as furniture movement?" To address this research question, we refined the basic method so that it periodically updates the classification model and adapts to changes. The core concept is that the system performs *periodic checks*, which means it periodically pings known-label devices (those whose inside/outside state is known) and collects their Wi-Fi response frames. Abedi et al. [1] demonstrate that even non-responsive or not-on-network Wi-Fi devices acknowledge incoming frames and respond with ACK signals, enabling this approach for a broad range of target devices, without hardware or software modification. These responses, processed as described in Section 4.1.3, are then incorporated into the training dataset to retrain the model, so that the model can account for any environment-induced changes in the response frames. By default, our system conducts periodic checks at 3 A.M. daily, which for most homes coincides with a period of reduced smart-home device activity and minimal movement. We elaborate on the whole process below.

5.1 Method

This section describes the design of the periodic adaptation method.

5.1.1 Training Data Collection. We define the data collected in the initial classification model training phase, described in Section 4.1.3, as the **initial training data**. Any new training data collected after the initial model training phase are defined as the **supplementary training data**. The supplementary training data may contain data from anchor devices (which may be labeled as either inside or outside), known-inside, and known-outside devices. We collectively refer to these devices as *known-label* devices due to their pre-labeled location status. During periodic adaptation, the system pings each known-label device 100 times to gather their corresponding Wi-Fi response frames. The result is a fresh set of labeled training data, which can be merged with previously collected training data to train an updated model.

5.1.2 Model Updating. As before, the hub computes a sliding window with a window size of $\omega = 10$ frames and a step size of 5 to extract aggregated features for each device's data. While the supplementary data provides more recent insights, the initial training dataset may encompass locations not represented in the supplementary set. Although the home's radio environment may change due to the relocation of movable items, such as furniture, we assume the fundamental structure of the residence does not change. The initial training dataset, collected manually by walking around the home, still contains information that reflects the areas that are not covered by the anchor or known-label devices. Therefore, the initial training dataset should be lessened in the training dataset instead of being simply removed.

We propose the following method for merging and balancing the initial training data with the subsequent daily supplementary training data. This approach ensures that the training data used for model retraining incorporates historical information and the latest updates in an appropriate ratio. The formula for calculating the latest training data T_i for day i in periodic adaptation is:

$$T_i = A_i \cup B_i \tag{4}$$

where

$$A_i = \{x \mid x \in S_i, \text{ selected with probability } \alpha\} \tag{5}$$

$$B_i = \{x \mid x \in T_{i-1}, \text{ selected with probability } (1-\alpha)\} \tag{6}$$

where T_i is the training data for periodic adaptation on day i, S_i is the supplementary training data collected on day i, T_{i-1} is training data for day $i-1$, and α is a weighting factor ($0 < \alpha < 1$).

For the first day when $i = 1$, T_0 represents the initial training data collected through the walk-around process as described in Section 4.1.3. S_1 represents the data collected during the first day of operation as part of the first periodic adaptation operation.

This formula incorporates a weighting factor α that balances the influence of the prior day's training data (leveraging previous home information), and the latest day's supplementary training data (reflecting recent changes). We randomly select a fraction $(1-\alpha)$ of data points from the previous day's training data T_{i-1} and combine them with a randomly selected fraction α of the supplementary data S_1 collected on the current day i. This combined data T_i undergoes resampling using the SMOTE oversampler, which is the same as described in Section 4.1.3, to address inside/outside dataset imbalance. Then, the resampled T_i is used to retrain the classification model, enabling it to adapt to the latest home environment change.

5.2 Evaluation of Periodic Adaptation

We conducted a ten-day experiment to evaluate the performance of periodic adaptation.

5.2.1 Experiment Design. Our ten-day experiment enabled us to observe the system's performance under diverse scenarios, including typical daily activities and deliberate actions intended to modify the indoor environment that necessitates model updates.

We deployed our system and smart devices, as described in Table 4; device categories are defined in Section 3.1. We conducted this experiment in a new home, which allowed us to conduct a ten-day experiment and change the interior as needed. Home 4 occupies one-half of the second floor of a single house, with other homes and residents on the first floor and the other side of the second floor. There were unknown devices around the experimental home, within the sniffers' Wi-Fi detectable range.

Table 4. The devices used for evaluating periodic adaptation, and their categories.

Category		Devices
Known device	**Anchor**	Inside: Observers × 8, TV, router, smart switch × 2 Outside: smart switch
	Known-inside	Smartphone, tablet, laptop, Google home hub
	Known-outside	Smart camera
	Known-other	Smartphone, laptop, Raspberry Pi
Unknown device		Neighbors' or passersby's equipment

Below, we listed the activities of the home residents during this ten-day experiment. "Living normally" involves the resident's typical daily routines, including walking around the home, cooking, sleeping, moving small objects, and periods spent outside of home – in short, the residents go about their daily lives.

Day 0: Hardware setup, initial training data collection, and model training, as described in Section 4.1.3, and the system was activated.

Day 1-4: The home resident lived in the home normally.

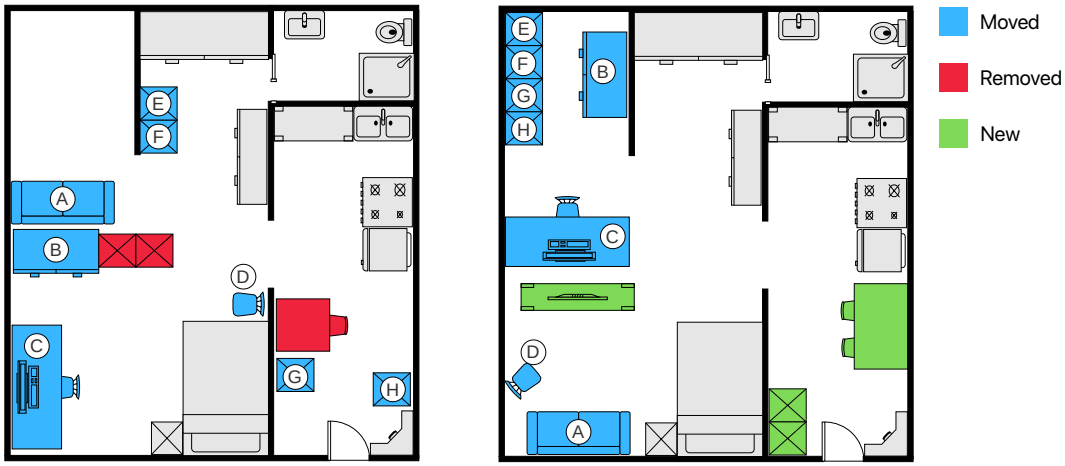

Fig. 9. Floor plan of Home 4, and positions of objects before and after intentional rearrangements on Day 5. Objects that were changed position in the home are marked in blue and lettered to show where they moved. Objects removed from the home are marked in red. New additions that were moved into the home are marked in green. The router icons show the distribution of observers in Home 4.

Day 5: To simulate a significant environmental change, the home resident added, removed, and shifted the locations of various large items in the home from 9 to 11 AM. Figure 9 shows the floor plan of Home 4 and the furniture and objects in it. We use labels to show where the objects were before and after they were purposely moved, and colors to indicate items added or removed. Table 5 provides a detailed description of the objects and how they were moved in this experiment. During the remaining hours of Day 5, the resident lived normally at home.

Day 6-10: The home resident lived in the home normally.

To comprehensively evaluate the effectiveness of periodic adaptation, we collected additional data for evaluation purposes only. Following each daily model update (scheduled for 3 AM), the system collected the first 100 Wi-Fi frames for each known-label device (including anchor, known-inside, and known-outside devices) at hourly intervals from 3 AM to 11 PM. This hourly collected data served as a test dataset, to calculate the classification accuracy for each hour. Subsequently, these hourly accuracies were averaged to get a whole-day average classification accuracy for all devices, providing a more thorough evaluation of the periodic adaptation's performance. It is important to note that such frequent, hourly evaluation data collection is unnecessary in a real-world deployment scenario.

To determine the optimal α for the periodic adaptation method, we evaluated its performance with various α values. We observed that the system achieved the best average balanced accuracy over this ten-day experiment when $\alpha = 0.3$. Therefore, the results presented in the subsequent section are based on $alpha = 0.3$.

5.2.2 Analysis Methods. Although we developed periodic adaptation to address the challenge of environmental changes for long-term usage (**RQ2**), it also has the potential to diminish the need for manual effort during system initialization. As a result, we evaluated the periodic adaptation from two aspects.

First, we evaluated the robustness of the periodic adaptation method in the face of in-home environment changes, i.e., we evaluated the impact of the periodic adaptation approach on the accuracy of the model based on the model trained by manually training data collection, as mentioned in Section 5.1.1. Second, we evaluated

Table 5. The objects in the home that were purposely moved and how they were moved.

Operations	Objects
Changed position in the home (blue)	Couch (A), wardrobe (B), computer workstation (C), chair (D), and four storage boxes filled with clothes and miscellaneous items (E, F, G, H)
Removed from home (red)	Two delivery boxes for garbage, and a small table with a chair.
Moved into the home (green)	Two storage boxes for food, a large table with two chairs, a TV, and a TV stand.

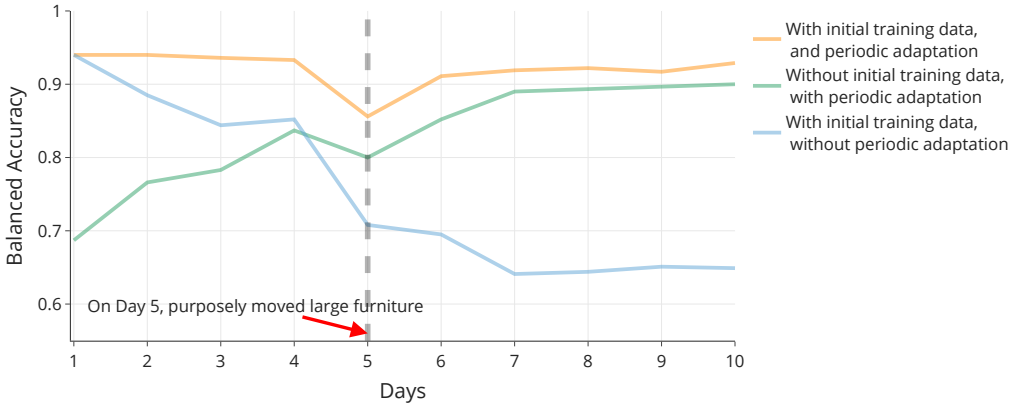

Fig. 10. This figure presents the balanced classification accuracy as it changed over 10 days. The x-axis denotes the day of the experiment, ranging from 1 to 10. The y-axis represents the balanced accuracy averaged across the day and across all known-label smart devices involved in this evaluation.

an approach that skips the initial training dataset – instead asking the user to label a few stationary devices as 'inside' or 'outside' – and gradually accumulates the dataset for training the model using only the periodic adaptation method; MOAT was able to learn an accurate model and adapt to indoor environmental changes.

5.2.3 *Results.*

With initial training data. Figure 10 shows how the balanced accuracy changed through the ten-day experiment. The graph clearly shows degraded model performance on Day 5, when the environment was purposely changed; this was true both with and without periodic adaptation throughout the whole day. However, with periodic adaptation, the system exhibited a quick recovery on Day 6 due to the incorporation of new information from the supplementary training data collected from known-label devices. On Day 10, with periodic adaptation, the balanced accuracy reached 92.9%. Without periodic adaptation, the balanced accuracy was only 64.9%. Supplementary training data provides valuable insights into the modified environment, allowing the model to adapt and regain performance.

Figure 10 also shows that the performance of the basic (non-adaptive) model degraded slowly even before Day 5, further emphasizing the need for adaptation. On Day 5, the performance with periodic adaptation still showed a better result than the method with no periodic adaptation.

The performance after the indoor environmental change shows that the periodic adaptation method was successful, addressing RQ2.

***Without** initial training data.* Given the strong performance of periodic adaptation, one might ask whether the initial training phase is needed at all? To address this, we evaluated the system performance without the initial training data (T_0). In this scenario, the system collects data from devices periodically and trains a classification model solely with data collected in periodic adaptation, i.e., using *only* the supplementary training data. Other than that, the data-processing and model-retraining methods remain the same, as stated above, for periodic adaptation. The system collects data on Day 1 and begins to train its first model after accumulating sufficient data, specifically after 3 AM, in line with our proposed default periodic adaptation schedule.

Figure 10 shows that even without the initial training dataset, the system starts Day 1 with a relatively lower accuracy of 68.7%, but it slowly improves over time. On Day 10, the method with no initial training data but with periodic adaptation finally reached an overall balanced accuracy of 90%, which is comparable to the system performance when initial training data is included.

For this approach to work, the system must be provided with MAC addresses for (at least some of) the 'known-label' devices from the home residents or users, which requires little manual effort. With an appropriate user interface, the installer may be able to identify those devices from an initial Wi-Fi scan after the hub and observers are installed. This action would be easier than walking around the home, inside and outside, to collect the initial training data. Also, since the periodic adaptation method constantly updates its training dataset, the new supplemental training data gradually replaces the initial training data. We envision that, without initial training data but with periodic adaptation, the system's classification accuracy will eventually come closer to that of the system with initial training data and periodic adaptation.

5.3 Discussion of RQ2

Through the evaluations for the periodic adaptation method, we have effectively addressed ***RQ2***. This method provides a solution that overcomes limitations identified in earlier discussions on the basic Moat system in Section 4.5.

The initial training data is meaningful to some extent, because when manually collecting training data, the transmitter device can cover the areas that anchors and known-label devices cannot cover. However, as the indoor environment changes over time, the accuracy of initial training data decreases. Traditional fingerprinting-based techniques depend on static environmental fingerprints, which can quickly become outdated as the environmental conditions change. The periodic adaptation method compensates for change by periodically updating the system to reflect current conditions, thereby preserving system accuracy and reliability even as the indoor environment evolves.

Furthermore, the periodic adaptation method reduces the need for manual collection of initial training data, a common hurdle in deploying a fingerprint-based system. This approach extends the basic Moat system to learn from anchors and known-label devices instead of solely on initial manually collected fingerprinting data. Without the initial training data, Moat takes a few days to accumulate training data and initialize. Similarly, Moat requires a transient period to adjust to furniture rearrangements, which may temporarily affect user experience.

In typical home settings, however, major rearrangements (such as moving furniture) are infrequent, thus minimizing the impact on system performance. Also, future work can explore strategies to further mitigate adaptation delays, such as increasing update frequency when a significant accuracy drop is detected. Additionally, our evaluations in Section 5.2 under normal living conditions, with disturbances like people moving and minor movements of objects, confirm that Moat maintains robust performance, ensuring that the daily activities of residents are unlikely to cause noticeable disruptions. Moreover, while many modern smart devices randomize their MAC addresses, this does not affect Moat's ability to detect whether a device is inside or outside a home, as Moat operates independently of MAC address identification.

Future work on refining the periodic adaptation method may include a longer-term evaluation of the system, extending beyond the initial ten-day assessment to provide a more comprehensive analysis of performance stability over time. We set the parameter α in Equation 5 to 30%, which was based on the outcomes we obtained in our experiments. In real-world applications, however, we envision there should be more automated tests to adjust α to suit specific environments. Additionally, a user study investigating typical patterns of furniture rearrangements in homes could offer insights into how frequently environmental reconfigurations occur and their consequent impact on system performance. This research could also explore user perceptions of accuracy fluctuations and their willingness to rely on Moat in everyday settings.

6 Limitations and Future Work

In this section, we discuss the limitations of the system design and evaluation, and opportunities for future work.

6.1 Limitations

Our current device discovery system only monitors Wi-Fi-based devices. Devices based on other protocols, such as Bluetooth or ZigBee, are not monitored by our system, but the concepts discussed above could be adapted for them.

6.1.1 Number of Observers. In our evaluation for the number of observers (Section 4.3.5), we proved that with only a small number of observers, the basic inside/outside detection system is already able to achieve good classification accuracy. Specifically, with 3 observers installed, the average balanced accuracy of the three homes was around 90%. This number of observers should not be a major burden for home residents. The evaluation also shows that, however, with increasing numbers of observers, the classification accuracy also increases (but by a smaller amount).

While a large number of observers can enhance classification accuracy, it may be impractical or burdensome for residents. Conversely, fewer observers can simplify deployment but compromise performance. We also acknowledge that some people may not want extra pieces of hardware, such as observers, in their homes. Building an inside/outside detection system without introducing extra hardware, or diminishing the amount of additional hardware, requires future work.

6.1.2 Position of Observers. The performance of this system relies on well-positioned observers. Although we experimentally explored observer placement in Section 4.3.5, further work is needed to develop mechanisms to recommend and optimize observer placement. Furthermore, in a real home, a resident may need (or want) to relocate observers (e.g., to use the electrical outlet for another purpose). Even repositioning a single observer necessitates model retraining. Observer relocation might initially lead to decreased performance, but the periodic adaptation mechanism would gradually recover performance by incorporating new supplementary training data obtained from the adjusted observer setup. Evaluating the system's resilience to observer relocation and exploring strategies for adapting the model in such scenarios are potential areas for future investigation.

6.1.3 Wi-Fi Channel. Another limitation of our current prototype is that the observers scan only one Wi-Fi channel, the same channel used by the home's network. This limitation becomes particularly relevant in scenarios when adversaries employ separate Wi-Fi networks for covert communication and data collection of the home. Consider an adversary that installs indoor sensor devices to spy on the home, leveraging a separate Wi-Fi network for coordination and data collection. Since the observers can only scan one channel at a time, coordinated multi-channel observation is crucial for comprehensive monitoring. Deshpande did some preliminary work on this question, but not in the inside/outside context [7].

6.1.4 Experiment Design. We conducted experiments in three residences, but future work should explore more homes of various sizes and types. None of the homes we evaluated are large houses or apartments (more than 250 square meters). We did not need many observers (or known-label devices for periodic adaptation) to cover most areas of the home. In a large home, there may be spaces that are not 'covered' by the radio range of observers or known-label devices; the classification accuracy may be low in such areas. Furthermore, in our periodic adaptation experiments, we employed enough known-label devices to 'cover' all the areas and provide enough information for periodic adaptation. In reality, there may be less known-label devices, and the coverage can become an issue.

6.2 Future Work

There are several opportunities for future work to extend the evaluation of our approach and to explore enhancements.

6.2.1 Optimization of Observers. In Section 4.3.5, we explore the effect of the number and locations of observers. Although we obtained reasonable accuracy with only four observers, questions remain. For example, what method can suggest a suitable number of observers for a home of a given size and layout? What automated tools could guide the user in optimizing the placement of observers? How can observer placement improve accuracy near the corners and boundaries of the home? More work is required to answer these questions properly.

6.2.2 Sophisticated Adversary. A sophisticated adversary may use a modified Wi-Fi device to modify the physical characteristics (signal strength, phase, direction) of its Wi-Fi transmissions to deceive our system, attempting to make it classify its inside devices as 'outside', or outside devices as 'inside.' Given the dispersed nature of the observers, it would be extremely difficult for an adversary to properly shape transmissions in a way that would mislead the model's classification, but we have not studied this possibility in detail.

6.2.3 User Study. Further studies are essential for future development of Moat. First, a study to explore users' perceptions of the ease of installing and training such a system would provide valuable insights to guide system improvement. Second, the manner in which Moat communicates and notifies its discoveries to users requires careful studies. Key aspects to study may include the information users want in these notifications, the frequency of notifications, the selection of devices on which to notify the user, and the interface for conveying information to the user. All these aspects are worthy of further exploration, and require comprehensive user studies to ensure the system meets users' needs effectively.

6.2.4 User Interface of Moat. A user-friendly interface and visualization platform would be necessary for users to understand their smart-home network. We envision a website or an application that displays the smart devices the end-user has and whether they are inside or outside the home. The interface needs intuitive means to notify the user of new devices or unusual behavior (an anchor device that moved or an inside device that went outside unexpectedly) so the users can make informed decisions and take action accordingly.

6.2.5 Moat and HCI. We envision Moat can also enhance the user experience by providing an opportunity to better support context awareness for IoT systems than other practices that do not provide device inside/outside status information. For instance, one may throw a party in their yard and move their smart lights and speakers to the outdoor environment. Moat can automatically detect the change, setting the lights' brightness and the speakers' volume to one that is more proper for outdoor use. It can also remind the resident if some of those devices are left outdoors when the party ends. As another example, when the user is inside the home Moat can help the user's smartphone to prioritize access to apps and services that are frequently used indoors, such as smart TVs or thermostats, reducing the effort of searching for the right device that they may have at home. As with security and privacy, Moat provides the technological foundation for user-interface services and innovations.

7 Conclusion

Device discovery is critical to managing smart-home networks; the ability to identify devices within or beyond the physical confines of a home can greatly aid its residents in monitoring and managing their devices. To this end, we proposed Moat, a system that passively sniffs Wi-Fi transmissions from all smart devices in and around a home and uses the physical properties of these transmissions to differentiate devices located inside and outside of the home – according to the residents' own notion of 'inside' and 'outside.' We tested our system in three homes and achieved an overall accuracy of 95.6%. We evaluated the system from different perspectives, including the performance of different classifiers, accuracy for different categories of target devices, the impact of observer locations and numbers, and accuracy for moving targets. We then extended the method to enable it to adapt to slow or sudden changes in the home's physical environment, maintaining strong performance up to 93%. Based on the periodic adaptation method, we proposed a method to initialize the system without manual effort in model training and achieved over 90% accuracy. We demonstrated that this adaptive method allows the system to learn over time without requiring the home resident to collect data for training.

Acknowledgments

We gratefully acknowledge the early contributions of former students Paul Gralla and KaiYao Weng, as well as Dr. Jared D. Chandler for his valuable assistance during the revision process. This research results from the SPLICE research project, supported by a collaborative award from the NSF SaTC Frontiers program under award number CNS-1955805, and the VeChain Foundation. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the sponsors. Any mention of specific companies or products does not imply any endorsement by the authors, by their employers, or by the sponsors.

References

[1] Ali Abedi and Omid Abari. 2020. WiFi Says" Hi!" Back to Strangers!. In *Proceedings of the ACM Workshop on Hot Topics in Networks (HotNets)*. 132–138.

[2] Ali Abedi and Deepak Vasisht. 2022. Non-cooperative Wi-Fi localization & its privacy implications. In *Proceedings of the Annual International Conference on Mobile Computing And Networking (MobiCom)*. ACM, 570–582.

[3] Nico Castelli, Corinna Ogonowski, Timo Jakobi, Martin Stein, Gunnar Stevens, and Volker Wulf. 2017. What Happened in my Home?: An End-User Development Approach for Smart Home Data Visualization. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*. ACM, 853–866.

[4] George Chalhoub, Martin J. Kraemer, Norbert Nthala, and Ivan Flechais. 2021. It did not give me an option to decline: A Longitudinal Analysis of the User Experience of Security and Privacy in Smart Home Products. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*. ACM, 555:1–555:16.

[5] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* 16 (2002), 321–357.

[6] Ionut Constandache, Romit Roy Choudhury, and Injong Rhee. 2010. Towards mobile phone localization without war-driving. In *2010 Proceedings IEEE INFOCOM*. IEEE, 1–9.

[7] Udayan Deshpande. 2008. *A Dynamically Refocusable Sampling Infrastructure for 802.11 Networks*. Ph.D. Dissertation. Dartmouth College Computer Science, Hanover, NH. https://www.cs.dartmouth.edu/~kotz/research/deshpande-thesis/index.html Available as Dartmouth Computer Science Technical Report TR2008-620.

[8] Rizanne Elbakly and Moustafa Youssef. 2016. A Robust Zero-Calibration RF-Based Localization System for Realistic Environments. In *International Conference on Sensing, Communication, and Networking (SECON)*. IEEE, 1–9.

[9] Pooyan Shams Farahsari, Amirhossein Farahzadi, Javad Rezazadeh, and Alireza Bagheri. 2022. A Survey on Indoor Positioning Systems for IoT-Based Applications. *IEEE Internet of Things Journal* 9, 10 (2022), 7680–7699.

[10] Harald T Friis. 1946. A note on a simple transmission formula. *Proceedings of the IRE* 34, 5 (1946), 254–256.

[11] Sidney Fussell. 2019. Airbnb Has a Hidden-Camera Problem. https://www.theatlantic.com/technology/archive/2019/03/what-happens-when-you-find-cameras-your-airbnb/585007/.

[12] Francesco Gringoli, Matthias Schulz, Jakob Link, and Matthias Hollick. 2019. Free your CSI: A channel state information extraction platform for modern Wi-Fi chipsets. In *Proceedings of the International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization (WiNTECH)*. 21–28.
[13] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. 2008. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*. IEEE, 1322–1328.
[14] Humboldt-Universität. 2017. Packet transmission time in 802.11. https://sarwiki.informatik.hu-berlin.de/Packet_transmission_time_in_802.11.
[15] Mohamed Ibrahim, Hansi Liu, Minitha Jawahar, Viet Nguyen, Marco Gruteser, Richard Howard, Bo Yu, and Fan Bai. 2018. Verification: Accuracy Evaluation of WiFi Fine Time Measurements on an Open Platform. In *Proceedings of the International Conference on Mobile Computing and Networking (MobiCom)*. ACM, 417–427.
[16] IEEE Standards Association. 2016. IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.
[17] IPX1031. 2019. Survey: Do Airbnb Guests Trust Their Hosts? https://www.ipx1031.com/airbnb-guests-trust-hosts/.
[18] Timo Jakobi, Gunnar Stevens, Nico Castelli, Corinna Ogonowski, Florian Schaub, Nils Vindice, Dave W. Randall, Peter Tolmie, and Volker Wulf. 2018. Evolving Needs in IoT Control and Accountability: A Longitudinal Study on Smart Home Intelligibility. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)* 2, 4 (2018), 171:1–171:28.
[19] David Janssen. 2023. Many Airbnbs have cameras installed, especially in the US, Canada and Singapore. https://vpnoverview.com/news/camera-presence-airbnb-accommodations/.
[20] Sophie Jeong and James Griffiths. 2019. Hundreds of motel guests were secretly filmed and live-streamed online. https://edition.cnn.com/2019/03/20/asia/south-korea-hotel-spy-cam-intl/.
[21] Haojian Jin, Boyuan Guo, Rituparna Roychoudhury, Yaxing Yao, Swarun Kumar, Yuvraj Agarwal, and Jason I. Hong. 2022. Exploring the Needs of Users for Supporting Privacy-Protective Behaviors in Smart Homes. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*. ACM, 449:1–449:19.
[22] Haojian Jin, Gram Liu, David Hwang, Swarun Kumar, Yuvraj Agarwal, and Jason I. Hong. 2022. Peekaboo: A Hub-Based Approach to Enable Transparency in Data Processing within Smart Homes. In *Symposium on Security and Privacy*. IEEE, 303–320.
[23] Palanivel A. Kodeswaran, Ravi Kokku, Sayandeep Sen, and Mudhakar Srivatsa. 2016. Idea: A System for Efficient Failure Management in Smart IoT Environments. In *Proceedings of the Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*. ACM, 43–56.
[24] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. 2015. SpotFi: Decimeter Level Localization Using WiFi. In *Proceedings of the ACM Conference on Special Interest Group on Data Communication*. ACM, New York, NY, USA, 269–282.
[25] Fan Li, Chunshui Zhao, Guanzhong Ding, Jian Gong, Chenxing Liu, and Feng Zhao. 2012. A reliable and accurate indoor localization method using phone inertial sensors. In *Proceedings of the 2012 ACM conference on ubiquitous computing*. 421–430.
[26] Liqun Li, Pan Hu, Chunyi Peng, Guobin Shen, and Feng Zhao. 2014. Epsilon: A visible light based positioning system. In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*. 331–343.
[27] Liqun Li, Guobin Shen, Chunshui Zhao, Thomas Moscibroda, Jyh-Han Lin, and Feng Zhao. 2014. Experiencing and handling the diversity in data density and environmental locality in an indoor positioning service. In *Proceedings of the 20th annual international conference on Mobile computing and networking*. 459–470.
[28] Mo Li, Pengfei Zhou, Yuanqing Zheng, Zhenjiang Li, and Guobin Shen. 2014. IODetector: A generic service for indoor/outdoor detection. *ACM Transactions on Sensor Networks (TOSN)* 11, 2 (2014), 1–29.
[29] Junjie Liu. 2014. Survey of wireless based indoor localization technologies. *Department of Science & Engineering, Washington University* (2014).
[30] Jia Liu, Atilla Eryilmaz, Ness B. Shroff, and Elizabeth S. Bentley. 2017. Understanding the Impacts of Limited Channel State Information on Massive MIMO Cellular Network Optimization. *Journal on Selected Areas in Communications* 35, 8 (2017), 1715–1727.
[31] Alex T. Mariakakis, Souvik Sen, Jeongkeun Lee, and Kyu-Han Kim. 2014. SAIL: single access point-based indoor localization. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys)*. ACM, 315–328. https://doi.org/10.1145/2594368.2594393
[32] Pardis E Naeini, Sruti Bhagavatula, Hana Habib, Martin Degeling, Lujo Bauer, Lorrie Cranor, and Norman Sadeh. 2017. Privacy Expectations and Preferences in an IoT World. In *Symposium on Usable Privacy and Security (SOUPS)*. USENIX, 399–412.
[33] Joan Palacios, Paolo Casari, and Joerg Widmer. 2017. JADE: Zero-knowledge device localization and environment mapping for millimeter wave systems. In *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 1–9.
[34] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[35] Gregary B Prince and Thomas DC Little. 2012. A two phase hybrid RSS/AoA algorithm for indoor device localization using visible light. In *2012 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 3347–3352.
[36] Valentin Radu, Panagiota Katsikouli, Rik Sarkar, and Mahesh K Marina. 2014. A semi-supervised learning approach for robust indoor-outdoor detection with smartphones. In *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*. 280–294.
[37] Niranjini Rajagopal, Patrick Lazik, Nuno Pereira, Sindhura Chayapathy, Bruno Sinopoli, and Anthony Rowe. 2018. Enhancing indoor smartphone location acquisition using floor plans. In *2018 17th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE, 278–289.
[38] Matthias Schulz, Daniel Wegemer, and Matthias Hollick. 2017. Nexmon: The C-based Firmware Patching Framework. https://nexmon.org
[39] Moustafa Seifeldin, Ahmed Saeed, Ahmed E Kosba, Amr El-Keyi, and Moustafa Youssef. 2012. Nuzzer: A large-scale device-free passive localization system for wireless environments. *IEEE Transactions on Mobile Computing* 12, 7 (2012), 1321–1334.
[40] Souvik Sen, Romit Roy Choudhury, Bozidar Radunovic, and Tom Minka. 2011. Precise indoor localization using PHY layer information. In *Proceedings of the 10th ACM Workshop on hot topics in networks*. 1–6.
[41] William Seymour, Martin J. Kraemer, Reuben Binns, and Max Van Kleek. 2020. Informing the Design of Privacy-Empowering Tools for the Connected Home. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*. ACM, 1–14.
[42] Rahul Anand Sharma, Elahe Soltanaghaei, Anthony Rowe, and Vyas Sekar. 2022. Lumos: Identifying and Localizing Diverse Hidden IoT Devices in an Unfamiliar Environment. In *USENIX Security Symposium*. 1095–1112.
[43] Vandit Sharma and Mainack Mondal. 2022. Understanding and Improving Usability of Data Dashboards for Simplified Privacy Control of Voice Assistant Data. In *Proceedings of USENIX Security Symposium*. USENIX, 3379–3395.
[44] Akash Deep Singh, Luis Garcia, Joseph Noor, and Mani Srivastava. 2021. I always feel like somebody's sensing me! A framework to detect, identify, and localize clandestine wireless sensors. *Proceedings of the USENIX Security Symposium* (2021), 1829–1846.
[45] Navneet Singh, Sangho Choe, and Rajiv Punmiya. 2021. Machine Learning Based Indoor Localization Using Wi-Fi RSSI Fingerprints: An Overview. *IEEE Access* 9 (2021), 127150–127174.
[46] Elahe Soltanaghaei, Avinash Kalyanaraman, and Kamin Whitehouse. 2018. Multipath triangulation: Decimeter-level wifi localization and orientation with a single unaided receiver. In *Proceedings of the 16th annual international conference on mobile systems, applications, and services*. 376–388.
[47] Biljana L Risteska Stojkoska and Kire V Trivodaliev. 2017. A review of Internet of Things for smart home: Challenges and solutions. *Journal of Cleaner Production* 140 (2017), 1454–1464.
[48] Yang Tian, Ruipeng Gao, Kaigui Bian, Fan Ye, Tao Wang, Yizhou Wang, and Xiaoming Li. 2014. Towards ubiquitous indoor localization service leveraging environmental physical features. In *IEEE infocom 2014-ieee conference on computer communications*. IEEE, 55–63.
[49] He Wang, Souvik Sen, Ahmed Elgohary, Moustafa Farid, Moustafa Youssef, and Romit Roy Choudhury. 2012. No need to war-drive: unsupervised indoor localization. In *International Conference on Mobile Systems, Applications, and Services (MobiSys)*. ACM, 197–210.
[50] Jiang Xiao, Kaishun Wu, Youwen Yi, Lu Wang, and Lionel M Ni. 2013. Pilot: Passive device-free indoor localization using channel state information. In *2013 IEEE 33rd International Conference on Distributed Computing Systems*. IEEE, 236–245.
[51] Jiang Xiao, Zimu Zhou, Youwen Yi, and Lionel M. Ni. 2016. A Survey on Wireless Indoor Localization from the Device Perspective. *Comput. Surveys* 49, 2 (Jun. 2016), 1–31. https://doi.org/10.1145/2933232
[52] Jie Xiong and Kyle Jamieson. 2013. ArrayTrack: A Fine-Grained Indoor Location System. In *Proceedings of the USENIX Symposium on Networked Systems Design and Implementation*. USENIX, 71–84.
[53] Jie Xiong, Karthikeyan Sundaresan, and Kyle Jamieson. 2015. Tonetrack: Leveraging frequency-agile radios for time-based indoor wireless localization. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. 537–549.
[54] Moustafa Youssef and Ashok K. Agrawala. 2005. The Horus WLAN location determination system. In *Proceedings of the International Conference on Mobile Systems, Applications, and Services (MobiSys)*. ACM, 205–218.
[55] Eric Zeng and Franziska Roesner. 2019. Understanding and improving security and privacy in multi-user smart homes: A design exploration and in-home user study. In *28th USENIX Security Symposium (USENIX Security 19)*. USENIX Association, Santa Clara, CA, 159–176. https://www.usenix.org/conference/usenixsecurity19/presentation/zeng

A Appendix: Network Communication Congestion

In this appendix we examine the network congestion on the communication channel between the observers and the hub. Recall that each observer receives a frame from a target device and sends CSI/RSSI measurements extracted from that frame to the hub on a separate Wi-Fi data channel. A congestion problem could arise on the data channel in environments where target devices transmit rapidly. To illustrate, consider a network where a target device transmits frames as fast as possible. If there are n observers, without a mitigation strategy, there would be n transmissions to the hub for each target device frame. If the target device channel is heavily utilized, the data channel may need to be as much as n times larger than the target device channel to accommodate the network traffic.

Moat mitigates this problem by aggregating the CSI/RSSI from 10 target device frames in a single data channel frame. To analyze this approach, we first examine the minimum time for the target device to send one frame and then estimate the minimum theoretical time for a target device to transmit 10 frames. Next we evaluate the minimum theoretical time for n observers to send aggregated data to the hub. If all observers are able to send data in less time than it takes the target device to send 10 frames, the data channel will be sufficient to handle the traffic. We see that by aggregating data, Moat observers eliminate a large portion of Wi-Fi overhead and are able to keep up with even the fastest target devices.

A.1 Target Device

When a device sends a Wi-Fi frame, it adds a preamble, header, and checksum to the data payload. The device then waits for acknowledgment from the access point before sending the next frame. We estimate the time for a target device to send a Wi-Fi frame in Table 6.

Table 6. Time for a target device to send one Wi-Fi frame.

Element	**Time** (microseconds) [14]
Frame preamble, header[2], checksum	282
Data time[3]	4
SIFS[4] + Ack[5]	44
Total for one target frame	330

Including the preamble, header, checksum, SIFS, and acknowledgment, the total time to transmit the smallest Wi-Fi frame is 330 microseconds. Frames with a larger data payload would take longer to send. Importantly, we see 326 of 330 microseconds is non-data overhead. By aggregating 10 target device frames into a single data channel frame, Moat eliminates the overhead of 9 frames.

Because a single frame takes 330 microseconds, the minimum theoretical time needed for a target device to send 10 frames is therefore 3,300 microseconds. This estimate ignores real-world considerations such as back off time when other devices transmit, resulting in channel usage inefficiencies. We also note that Wi-Fi uses Carrier Sense Multiple Access (CSMA) where only one device transmits at a time, so Moat does not need to accommodate multiple simultaneous transmitters.

[2]Assumes 54 Mbps for the header.
[3]Wi-Fi uses 4 microseconds to send one symbol, so the smallest data payload will take 4 microseconds to transmit.
[4]Short Interframe Space defined in the 802.11 specification [16].
[5]Time for the access point to transmit an acknowledgment.

A.2 Moat Observers

We model the time for each Moat observer to send data to the hub in Table 7. We first note that each observer measures CSI on 52 subcarriers, where each subcarrier's CSI is represented by a 2-byte integer, giving a total of 104 bytes of CSI that must be sent to the hub for each frame. Each observer also captures the RSSI and MAC address in 2 and 6 bytes respectively, for a total of 112 bytes to send per target device frame.

Moat compresses this data before sending it to the hub. In our experiments the mean compression ratio was 0.43, with a standard deviation of 0.0092. This reflects a moderately high level of compressibility with a relatively low variance in compression efficiency across different batches. In Table 7, we use a conservative compression ratio of 0.5 in our estimation, reducing the data to send to the hub from 112 bytes to 56 bytes or 448 bits.

Next we estimate the time to send the compressed 448 bits. We use MCS 6 (Modulation Coding Scheme) [16] on one 20 MHz channel. This scheme uses 64 QAM (6 bits per symbol) with a 3/4 coding scheme. An *OFDM symbol* is 64 subcarriers sending data simultaneously (but only 52 of those subcarriers carry data, the others are guard and pilot channels). Each OFDM symbol takes 4 microseconds in Wi-Fi, therefore to send 448 bits from 10 target frames' data over 52 data-carrying subcarriers using MCS 6 requires 20 OFDM symbols (see Table 7). These symbols take a total of 80 microseconds. After adding in the data frame's overhead (preamble, header, checksum, SIFS, and ACK) we estimate that takes an observer 406 microseconds to send the aggregated CSI/RSSI data to the hub.

Finally we can estimate the number of observers that can send their data to the hub before a target device sends another 10 frames. We previously estimated it takes the target device *at least* 3,300 microseconds to send 10 frames. Dividing this value by the 406 microseconds it takes the hub to send its 10-frame aggregated data suggests Moat can support 8.1 observers. We did not exceed 8 observers in our experiments.

We note, however, that these values are a rough sketch. In the real world, target devices will not transmit at the maximum theoretical speed at all times. Even if they did, the reduction in overhead by aggregating measurements, and compressing Moat frame content, allows Moat to handle the load.

Table 7. Calculation of time to send one aggregated data from an observer to the hub.

Element	Value	Units	Notes
MOAT data			
Subcarriers extracted	52	subcarriers	
Bytes/subcarrier	2	bytes	
Total CSI size	104	bytes	Number of subcarriers * byte/subcarrier
RSSI size	2	bytes	
MAC size	6	bytes	
Total MOAT data size	112	bytes	Total CSI size + RSSI size + MAC size
Compress data before sending			
Compression ratio	0.5	fraction	Assume conservative compression
Bytes after compression	56	bytes	Compression ratio * total data size
Bits after compression	448	bits	Bytes after compression * 8
Wi-Fi data time			
Bits/symbol	6	bits/symbol	BSPS=1, QPSK=2, 16QAM=4, 64QAM=6, 256QAM=8
Coding scheme	3/4	redundancy	2/3, 3/4, or 5/6
Number of symbols to send per frame	100	symbols	Roundup (bits/coding scheme/bits/symbol)
Total symbols to send	1000	symbols	Number of symbols to send per frame * 10 frames
Data subcarriers	52	subcarriers	
Number of OFDM symbols	20	OFDM symbols	Roundup (total symbols to send/data subcarriers)
Time per OFDM symbol	4.0	microseconds	
Data time	80	microseconds	Number OFDM symbols * (time per OFDM symbol)
Time for one observer			
Frame preamble, header, checksum	282	microseconds	Same as target device
Data time	80	microseconds	10 frames aggregated data
SIFS + Ack	44	microseconds	Same as target
Total time for one observer	406	microseconds	
Number of observers	8.1	observers	Target time (3300ms) / total time for one observer