ZeroTouch: Reinforcing RSS for Secure Geofencing

Nikola Antonijević
COSIC, KU Leuven
Leuven, Belgium
nikola.antonijevic@esat.kuleuven.be

Sayon Duttagupta
COSIC, KU Leuven
Leuven, Belgium
sayon.duttagupta@esat.kuleuven.be

Dave Singelée
COSIC, KU Leuven
Leuven, Belgium
dave.singelee@esat.kuleuven.be

Enrique Argones Rúa
COSIC, KU Leuven
Leuven, Belgium
enrique.argonesrua@esat.kuleuven.be

Bart Preneel
COSIC, KU Leuven
Leuven, Belgium
bart.preneel@esat.kuleuven.be

Abstract

Geofencing, the virtual demarcation of physical spaces, is widely used for managing the localisation of Internet of Things (IoT) devices. However, traditional localisation techniques face security challenges indoors due to signal interference and susceptibility to spoofing, often requiring extensive calibration or extra hardware, limiting scalability. In this work, we propose *ZeroTouch*, a machine learning-based system that leverages Received Signal Strength (RSS) measurements from multiple receivers to improve the security of geofencing without introducing additional deployment overhead. While RSS-based localisation is known to have inherent security limitations, we show that by aggregating RSS readings from multiple anchor points and detecting anomalies using an autoencoder model, *ZeroTouch* provides a practical and automated mechanism for verifying whether a device is inside or outside a defined boundary. Rather than serving as a standalone security mechanism, *ZeroTouch* enhances existing authentication frameworks by adding an additional *zero-touch* security layer that operates passively in the background. *ZeroTouch* eliminates manual calibration, removes the *human-in-the-loop* element, and simplifies deployment. We evaluate our solution in a realistic simulated environment and demonstrate that it achieves high accuracy in distinguishing between in-room and out-of-room devices, even in strong adversarial settings.

CCS Concepts

• **Security and privacy** → **Security services**; *Authorization*; Embedded systems security; • **General and reference** → Experimentation;

Keywords

IoT security; Device commissioning; Secure geofencing; RSS

ACM Reference Format:
Nikola Antonijević, Sayon Duttagupta, Dave Singelée, Enrique Argones Rúa, and Bart Preneel. 2025. ZeroTouch: Reinforcing RSS for Secure Geofencing. In *Proceedings of the 30th ACM Symposium on Access Control Models and Technologies (SACMAT '25), July 8–10, 2025, Stony Brook, NY, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3734436.3734448

SACMAT '25, Stony Brook, NY, USA
© 2025 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-1503-7/2025/07
https://doi.org/10.1145/3734436.3734448

1 Introduction

The Internet of Things (IoT) has transformed numerous sectors by enabling interconnected devices to collect, exchange, and act on data autonomously. As IoT expands into security-sensitive domains — such as healthcare facilities, industrial plants, and critical infrastructure — it becomes essential to enforce physical boundaries around secure areas [36]. In these scenarios, it is often not crucial to determine the exact location of a device but rather its presence within a predefined perimeter, such as a room or restricted area. This is known as geofencing [47], which aims to establish a virtual boundary and verify whether a device is inside or outside that boundary. Unlike precise localisation, geofencing usually only requires a binary determination (in/out) of a device's presence, simplifying implementation while increasing security in use cases where exact coordinates are unnecessary. Both indoor IoT localisation and geofencing have multiple applications, ranging from access control and device commissioning to resource allocation and monitoring.

Since Global Navigation Satellite Systems (GNSS[*]) solutions, such as GPS and Galileo, only work for outdoor localisation, alternative methods have been developed for indoor localisation. These are typically based on Time-of-Flight (ToF), Angle-of-Arrival (AoA), or Received Signal Strength (RSS) and rely on communication technologies such as Ultra-Wideband (UWB), Wi-Fi, or Bluetooth. Among these, RSS stands out due to its simplicity, compatibility with standard IoT devices, and ease of implementation. Moreover, it does not require additional hardware, making it particularly attractive for large-scale deployments. However, when measured at a single receiver, RSS has been shown to be susceptible to manipulation and environmental noise, making it unreliable for secure localisation or geofencing [3, 4, 13, 49].

In this paper, we introduce *ZeroTouch* and demonstrate that combining RSS data from multiple receivers enables anomaly detection in location claims, thereby providing an additional layer of security for geofencing. While an adversary may manipulate the RSS readings of a few devices, aggregating data from multiple receivers helps mitigate such attacks and adds an additional layer of defence, akin to a slice in the Swiss cheese model [15]. *ZeroTouch* is not intended to replace existing authentication methods but rather to complement them by offering a frictionless, zero-touch mechanism to enhance security. As a result, given that RSS is straightforward to obtain from receivers, we enhance the security of such systems

[*] https://www.gps.gov/systems/gnss/

by relying on RSS-based geofencing. Our work provides a cost-effective solution without requiring additional hardware, making it a practical and scalable approach for real-world deployments. More in detail, in this paper we present *ZeroTouch*, a geofencing system that enhances security by leveraging machine learning to mitigate the limitations of RSS for reliable perimeter-based presence detection. Our solution simplifies boundary verification while ensuring security against spoofing, making it highly suitable for IoT deployments where exact coordinates are not required. In our work, we use RSS of Wi-Fi as the metric to perform secure geofencing. While our model is implemented using Wi-Fi's RSS, the same approach could be extended to other technologies such as Bluetooth, LoRa, or Zigbee, as these protocols also provide RSS measurements. Since our methodology relies purely on RSS patterns rather than protocol-specific features, adapting it to different wireless technologies is straightforward, provided sufficient anchor points exist within the environment.

Contributions

In this paper, we present *ZeroTouch*, a novel and secure geofencing system that leverages Received Signal Strength (RSS) measurements and machine learning to verify whether a device is inside a defined perimeter. Our contributions are summarised as follows:

- We demonstrate that combining RSS data from multiple indoor receivers enables the detection of anomalies when a device outside a pre-defined geofence area attempts to spoof its location as being inside. This practical approach improves security by leveraging the aggregated behaviour of RSS measurements, making location spoofing more difficult for an adversary who has already bypassed initial authentication mechanisms.
- We design *ZeroTouch*, a secure and reliable mechanism to verify location claims within the context of geofencing. By using an autoencoder trained on legitimate RSS patterns, our approach requires no manual calibration or setup, reducing the *human-in-the-loop* factor to a minimum.
- We validate our approach in a well-defined and realistic threat model and simulation setting, testing multiple adversarial models, including attackers with varying transmission power levels. Our experiments show that *ZeroTouch* achieves more than 90% accuracy in distinguishing between inside and outside devices.
- We provide a scalable framework that allows users to balance accuracy and system usability by offering configurable levels of verification mechanisms, enabling seamless adaptability to diverse deployment scenarios.

Availability Statement. The entire source code and artefact underlying this paper can be found at https://github.com/KULeuven-COSIC/ZeroTouch.

2 Related Work

Various methods have been proposed to determine a device's location or its presence within a specific boundary in IoT systems. Techniques such as Time-of-Flight (ToF), Angle-of-Arrival (AoA), and Ultra-Wideband (UWB) have been widely studied and provide high accuracy. These methods typically require specialised hardware and extensive calibration, making them less practical for large-scale or dynamic IoT environments [34, 41, 42]. In contrast, Received Signal Strength (RSS)-based localisation does not rely on specialised hardware, making it attractive due to its simplicity and inherent compatibility with standard IoT devices, despite its limitations in accuracy and susceptibility to environmental interference [11, 19, 44, 48]. However, most existing RSS-based methods prioritise accuracy, reliability, and ease of use without addressing security considerations. Security-focused RSS localisation approaches, such as [28], primarily aim at mitigating the impact of malicious nodes already present within the network. Differently, our work targets the initial device verification and onboarding phase, establishing secure geofencing boundaries and effectively enhancing security during device commissioning.

Several works in *secure* localisation focus on precise location estimation using hardware-dependent methods or GPS modules, often requiring specialised equipment, trusted anchor nodes, or extensive calibration [10, 16, 25]. Other approaches employ machine learning for anomaly detection, such as one-class SVMs or gradient descent, but rely heavily on predefined anchor placements or high-quality labelled data, limiting scalability in dynamic environments [27, 30]. Furthermore, these works overlook security as a critical vector, lacking comprehensive threat models, security analyses, or evaluations against adversarial scenarios. Geofencing, which focuses on binary in/out determination, provides a scalable alternative to precise localisation in scenarios where exact coordinates are unnecessary. For example, geofencing has been used to restrict medical devices to specific rooms in hospitals, ensuring they operate only within authorised zones [1]. However, the current instantiation of geofencing by the Wi-Fi Alliance has been found to be insecure and vulnerable to distance manipulation attacks [38]. To the best of our knowledge, no existing solution proposes a secure geofencing framework or model that eliminates the need for additional hardware or calibration. *ZeroTouch* fills this gap by providing a scalable, hardware-independent, and machine-learning-driven approach to secure geofencing.

Furthermore, several other works have explored RSS-based solutions for device pairing and commissioning, aiming to reduce the *human-in-the-loop* element and improve usability. For instance, *Move2Auth* [50] and *SFIRE* [12] use RSS traces between an IoT device and a smartphone but require user gestures or movement, while *SenCS* [17] generates entropy from walking. Similarly, schemes like *Shake Well Before Use* [22] and *T2Pair* [20] rely on gestures or additional hardware, limiting practicality. Context-based approaches [2, 23] to prove proximity often suffer from security vulnerabilities or require extra equipment. A more straightforward method involves GNSS-based commissioning [21, 37], which is known to be ineffective indoors and susceptible to spoofing attacks [13, 26, 29, 43]. *ZeroTouch* addresses these shortcomings by leveraging inherently available RSS measurements from standard IoT antennas to enable secure and autonomous device commissioning. By aggregating RSS data from multiple receivers and employing machine learning to detect anomalies, *ZeroTouch* significantly mitigates location spoofing vulnerabilities associated with single-receiver RSS systems. Additionally, *ZeroTouch* avoids the need for additional hardware, calibration, or user interaction, making it a cost-effective, scalable, and practical geofencing solution.

3 System and Threat Model

This section first presents a comprehensive, high-level overview of our proposed model, highlighting the key components of our solution. Next, it outlines the threat model and the associated security assumptions.

3.1 System Model

In this paper we consider a geofence area as an indoor apartment with several rooms. However, our model is applicable to any indoor environment, including factory spaces, smart homes, or offices. The core objective of our scheme is to securely verify a prover's claim of being physically present within the secure area.

Our model is built upon a few fundamental assumptions. First, we assume that the rooms contain n pre-installed wireless devices, referred to as anchor nodes. These devices can be any IoT devices (e.g., smart lamps, thermostats, speakers, plugs, cameras) commonly found indoors. It is reasonable to expect that most indoor spaces will already have a number of these devices in place. We also assume these devices operate on Wi-Fi; however, as stated earlier, our solution applies to other wireless technologies as well, as it only relies on capturing the RSS. Next, they are connected to the network and maintain a secure connection with the central server, typically via a TLS-secured channel. This secure connection ensures that the communication between anchor nodes and the central server is protected from eavesdropping and tampering. It is important to note that we only need to be aware that anchor nodes are within the secure area, and we *do not* need to know their exact location. This eliminates the challenging task of precisely locating each anchor node, simplifying deployment and reducing setup complexity, thus making our model more practical for real-world applications. While implementation on existing IoT devices may require additional software capabilities (e.g., packet capture and RSS measurement), we assume such functionality is available. Exploring practical deployment across diverse IoT platforms is out of scope for this work.

To prove its presence within the area, a new wireless device (the prover) broadcasts a message, which is recorded by all anchor nodes. Each anchor node measures the RSS and forwards these measurements to the central server. The central server then aggregates the RSS measurements from the anchor nodes and employs our proposed verification algorithm to confirm the prover's presence within the geofence. Further details on the verification process are provided in Sect. 4.

3.2 Threat Model and Security Assumptions

ZeroTouch is designed to defend against adversaries attempting to bypass its boundary detection, allowing unauthorised devices to spoof their presence and gain access within a protected perimeter. Our threat model protects against external adversaries, that is, to protect against outside attackers/devices without physical access to the room. We assume a realistic adversary with the capability to manipulate Received Signal Strength (RSS) readings during the verification phase. We assume the training phase is secure and the enrolment and initialisation part is controlled. Specifically, the adversary is capable of adjusting its antenna transmission power, thus performing a *power-sweep attack*, to alter the RSS values observed by the anchor nodes within the boundary. While an even more powerful *ideal* adversary with unlimited capabilities (e.g., perfect directional antenna control, precise knowledge of all anchor positions, and complete understanding of the environment's geometry) would theoretically be impossible to defend against using only RSS, we opt to focus our analysis on more practical and realistic threats. Specifically, the adversary has access to sophisticated transmission equipment that allows control over the RSS values perceived by anchor nodes within the boundary. By manipulating transmission power, through signal amplification, the adversary attempts to recreate the RSS signature expected from an in-bound device. Our threat model also does *not* assume that the adversary is operating within the legal signal strength limits as defined by regulatory bodies such as the FCC in the United States[†] and under the RED in the European Union[‡] (which is typically restricted to 1 watt or 30 dBm). Additionally, the adversary has approximate knowledge of the anchor nodes' locations within the protected area, and is not restricted to a single place during the attack procedure.

To define the operational environment and the limitations under which ZeroTouch functions effectively, we make a few security and environmental assumptions. Firstly, we assume that the anchor nodes placed inside the room are tamperproof, and the adversary is working with an omnidirectional antenna from outside, trying to spoof their device into the insider network, as omnidirectional antennas are most commonly used in IoT deployments due to their uniform coverage and practicality [14, 33]. While directional antennas could theoretically provide attackers with more control, they require precise alignment with each receiver, which is impractical in dynamic environments and computationally challenging. Omnidirectional antennas are far more realistic for adversaries due to their ease of deployment and prevalence in IoT systems. Hence, we align with practical and realistic threat models while ensuring security.

4 ZeroTouch: High-level Overview

In this section, we explore the core elements of ZeroTouch. We begin by outlining the primary steps of our solution. Next, we discuss the machine learning aspect, specifically how the autoencoder is utilised to detect potential attacks. Finally, we also discuss the neural network baseline and how it compares to the autoencoder.

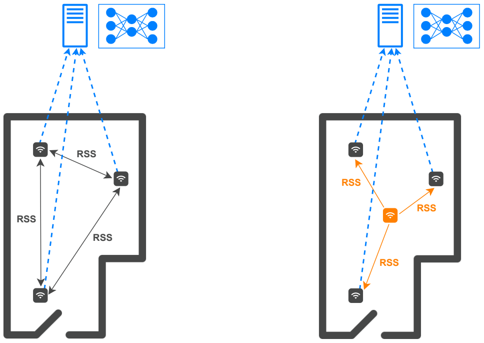
4.1 Protocol Flow

The protocol flow of ZeroTouch is structured into two main phases: the initialisation and verification phase. These phases define the necessary steps required to operate ZeroTouch.

Initialisation Phase. The first step of ZeroTouch is to capture the current model of a room, a process we refer to as the initialisation phase. Initially, the anchor nodes communicate with one another, transmitting standard messages and recording the corresponding RSS values to establish an RSS profile for the environment. Any message that includes the identity of an anchor node and can be verified as originating from it can be used to measure the RSS. Next, the RSS measurements from each anchor node are forwarded to the central server, which aggregates them into a complete dataset from various node positions. This process resembles an offline phase in

[†] https://www.ecfr.gov/current/title-47/chapter-I/subchapter-A/part-15

[‡] https://eur-lex.europa.eu/eli/dec_impl/2022/180/oj/eng



(a) Initialisation phase. **(b) Verification phase.**

Figure 1: Overview of the protocol flow in ZeroTouch, illustrating the two main phases: the initialisation phase and the verification phase.

wireless fingerprinting, where a database of RSS ‘fingerprints’ is created [40]. The final step in the initialisation phase is to utilise the measurements to train the machine learning model, ultimately producing a model of the room. For this purpose, we have chosen an autoencoder trained directly on the RSS measurements. Further details about the autoencoder are provided in Sect. 4.2. The training process is usually conducted only at the beginning; once the room model is established, retraining is unnecessary unless there are significant changes to the room's characteristics, such as major alterations in furniture arrangement or a drastic increase in the number of occupants. In such cases, rerunning the initialisation procedure is recommended. The initialisation phase is illustrated in Fig. 1a, where anchor nodes (depicted as grey wireless devices) communicate to measure RSS values. These measurements are then forwarded to the server (blue dashed line), which uses them to train the autoencoder and establish the room's model.

Verification Phase. The next step involves a new device (the prover) initiating the joining procedure by broadcasting a predetermined message to the anchor nodes, which includes the prover's ID. In the context of Wi-Fi, this message is analogous to probe requests or action frames sent by devices to announce their presence. As outlined in Sect. 3.1, the anchor nodes receive this message, measure the corresponding RSS, and forward the measurements to the central server. The central server then aggregates these measurements and processes them using the previously trained autoencoder. By comparing the reconstruction error to a predefined threshold, the server identifies whether the measurements originate from a device inside or outside the room. Specifically, if the reconstruction error exceeds the threshold, the measurements are classified as anomalous, indicating that the device attempting to prove its presence inside has failed and is thus determined to be outside. Further details on the autoencoder and the thresholding method used are provided in the following two subsections. The verification phase is illustrated in Fig. 1b, where a new device, depicted in orange (the prover), broadcasts a message that is received by the anchor nodes and used to determine the RSS. These measurements are then forwarded to the server, represented by the blue dashed lines, which uses the pre-trained autoencoder to verify the device's presence within the room.

4.2 Autoencoder-based anomaly detection

Before discussing the details of the autoencoder, let us first define the classification problem mathematically. It can be expressed as:

$$z = \mathbf{1}_{\{f(\mathbf{x})>\delta\}}, \tag{1}$$

where:

$\mathbf{x} = (RSS_1, RSS_2, \ldots, RSS_N)$ is the RSS-based vector representing the RSS values recorded by N anchor nodes.

$f(\mathbf{x})$ is a classifier function that assigns an anomaly score to the input $\mathbf{x}$, indicating its alignment with expected behaviour.

δ is the predefined threshold used to differentiate between inside and outside classifications.

z represents the final decision, where $z = 1$ denotes an anomaly (outside transmitter) and $z = 0$ indicates normal behaviour (inside transmitter).

The indicator function $\mathbf{1}_{\{f(\mathbf{x})>\delta\}}$ evaluates whether the anomaly score $f(\mathbf{x})$ is greater than the threshold δ. It is defined as:

$$\mathbf{1}_{\{f(\mathbf{x})>\delta\}} = \begin{cases} 1, & \text{if } f(\mathbf{x}) > \delta, \\ 0, & \text{if } f(\mathbf{x}) \leq \delta. \end{cases} \tag{2}$$

In this formulation, the anomaly score $f(\mathbf{x})$ is compared to δ, with the indicator function outputting 1 for outside classifications and 0 for inside classifications. This general formulation provides the foundation for designing a suitable classifier. Later, we show how an autoencoder effectively implements the function f.

An autoencoder is an artificial neural network designed for unsupervised learning, aimed at learning a good compressed representation of data [18]. It learns patterns and structures in unlabeled data without explicit target outputs. The autoencoder typically consists of two main components: the encoder and the decoder. The encoder compresses the input data into a lower-dimensional representation, known as the latent space. The decoder then takes this latent space, the compressed form of the input data, and attempts to reconstruct the original input as accurately as possible. Autoencoders have a variety of applications, one of which is anomaly detection.

The concept of using autoencoders for anomaly detection is well-established and has been applied in various fields [5]. In this approach, the autoencoder is first trained on normal, *non-anomalous* data, enabling it to reconstruct this data with minimal error, thereby minimising the reconstruction or residual error (the difference between the output and input data). When presented with anomalous data, the autoencoder typically fails to reconstruct it accurately, resulting in significantly higher residual errors compared to non-anomalous data. This behaviour demonstrates why the autoencoder is well-suited to serve as the function f defined earlier. The high reconstruction error indicates that the data deviates significantly from the expected (non-anomalous) patterns. Consequently, the data can be classified as an anomaly if the residual error exceeds a predefined threshold δ.

In our work, we train the autoencoder on a set of RSS measurements originating from anchor nodes located inside the room. In pattern recognition terminology, this approach is known as one-class classification, where our training set consists exclusively of one class — in this case, RSS measurements from indoor anchor nodes. Consequently, the autoencoder learns to accurately reconstruct RSS measurements characteristic of devices inside the room.

This approach reduces the need for collecting extensive datasets of potential anomalies, which may be impractical, incomplete, or infeasible. It allows us to rely *solely* on the anchor nodes already present within the room for training. When a legitimate device inside the room attempts to prove its presence, the reconstruction error is expected to remain low, similar to that observed during training. In contrast, when RSS measurements originate from a device outside the room, the autoencoder generates high reconstruction errors, facilitating their classification as anomalies (i.e., potential attacks). Although many variations of autoencoder neural networks exist, we demonstrate that employing a simple autoencoder with only a single hidden layer already yields effective results. Further details on the autoencoder used in our experiments are provided in Sect. 5.

Thresholding. Determining the threshold δ is a crucial component of the detection process, as it dictates whether a device is classified as genuine or anomalous. We aim to establish the threshold *a priori*, relying solely on the reconstruction errors observed during autoencoder training. This task is challenging, requiring defining a threshold using only one-class data [45]. Several methods can be used, and the approach we adopted is detailed in Sect. 6.1.

4.3 Baseline

Although our solution is based on an autoencoder, it is valuable to establish a theoretical baseline as a reference point for comparison. In our case, the baseline is a neural network using supervised learning, trained on RSS values obtained from both indoor and outdoor devices. This means the training set contains both classes, in contrast to the one-class training used in the autoencoder scenario. This baseline represents the best-case scenario achievable for this problem using neural networks. By comparing our solution to this theoretical baseline, we can assess its relative performance and observe the potential improvement that could be achieved if data from both classes were available. However, it is important to note that this baseline is not practical in real-world settings, as it requires data from both classes, which can be challenging to collect. Further information on the baseline, including its architecture and performance metrics, is provided in Appx. A.

5 Simulation Setup

In this section, we explore the details of the simulation setup of ZeroTouch, by outlining the specifics of the simulation environment. For this purpose, we utilised Wireless InSite [35] to model indoor radio wave propagation, allowing us to accurately simulate the RSS values within a particular indoor environment [35]. The generated data was subsequently analysed in MATLAB [39].

5.1 Wireless InSite

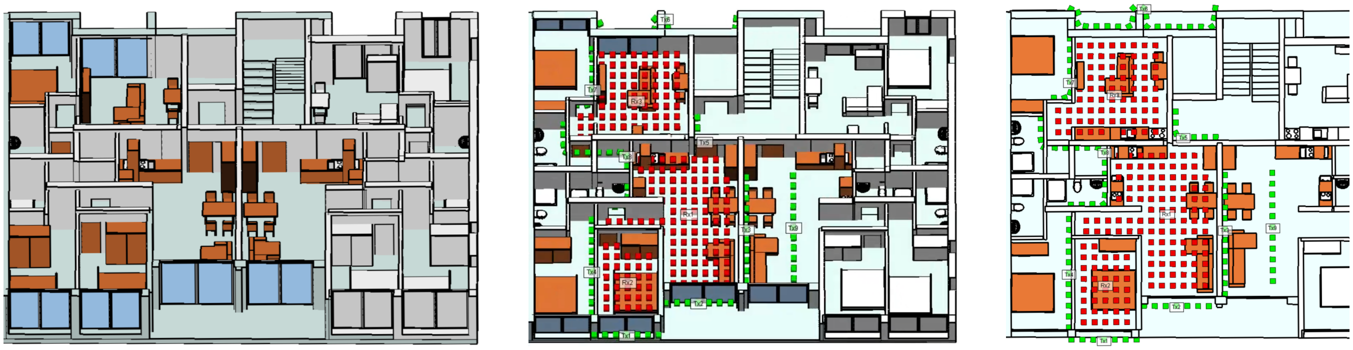
To evaluate a large number of transmitters and receivers, we opted to obtain our RSS measurements through the well-established simulation software Wireless InSite [35]. Wireless InSite was chosen for its advanced ray-tracing capabilities, which enable precise modelling and analysis of various indoor environments for wireless communication systems. It accounts for all relevant propagation effects, including reflection, diffraction, transmission, scattering, absorption, and the passage of signals through different surfaces and materials, to accurately predict wave propagation and RSS. Furthermore, academic studies have validated its results, demonstrating that it closely resembles real-life scenarios [24, 32]. The software supports both creating indoor environments and importing pre-existing models. To make our simulations as realistic as possible, we chose to import a complete architectural model of a floor in an apartment building, designed by a practising architect. This model, displayed in Fig. 2a, is part of a preliminary sketch draft from an urban design feasibility study conducted by architects at ZDL Studio [31]. The model includes multiple fully furnished rooms, incorporating a variety of materials: glass for windows, wood for furniture, drywall for interior walls, and concrete for exterior walls, floors, and ceilings.

Measurement points. We have defined the geofence area as a set of three rooms within the apartment, designating these rooms as a secure 'inside' region while all other areas of the apartment model are considered 'outside'. The three selected rooms are marked with red squares in Fig. 2b and Fig. 2c. Room one (Rx1) is the largest room, located in the middle; room two (Rx2) is the smallest, located at the bottom left; and room three (Rx3) completes the selection. These rooms represent a realistic selection, as they vary in size, shape, and furnishings. To obtain detailed RSS measurements, we utilised Wireless InSite's capability to set multiple measurement points and created a grid of transceivers within each room. Each red square represents a transceiver — a potential location for an anchor node or an inside prover — spaced 0.5 m apart, allowing us to collect measurements from various locations within each room, with a total of 223 potential inside locations. Thus, in our simulation, each red square can act as both a transmitter (in the case of a prover) and a receiver (in the case of an anchor node). For example, one red square can act as a transmitter while all other transceivers in the room serve as receivers, enabling us to measure the RSS (in dBm) from that particular transmitter at each location within the grid.

After defining the potential transceiver locations inside the geofence, we then establish the outside transmitter locations. These represent potential positions of an attacker attempting to falsely prove their presence within the secure geofence area. To maximise the attacker's advantage, we selected locations just outside the geofence boundaries, approximately 20 cm from the perimeter. These outside locations, marked as green squares in Fig. 2b and Fig. 2c, allow us to evaluate our solution under the worst-case conditions from a security point of view. Additionally, we included a row of potential transmitters on the right side, positioned away from the geofence boundary, for further testing. Altogether, these selected locations represent the possible attacker positions we aim to evaluate in our work, with a total of 102 outside transmitters.

Simulation parameters. The simulations were conducted under standard atmospheric pressure (1013.25 millibars). Wireless InSite offers various propagation models with different simulation approaches; for our purposes, the *X3D* model was selected, which simulates wireless transmission based on signal ray propagation [35]. Ray spacing was set to 0.25 degrees, with simulation parameters configured to include six reflections, three transmissions, and one diffraction, adequately representing the experimental scenarios [32]. Both transmitters and receivers were equipped with omnidirectional antennas operating at 2.4 GHz.

The computer used for our simulations is equipped with an i7-14700KF CPU, NVIDIA GeForce RTX 4080 16 GB GPU, and 32 GB of



(a) Apartment layout (ceiling not shown). **(b) Measurement points (side view).** **(c) Measurement points (top view).**

Figure 2: Apartment layout and measurement points. Fig. 2a depicts the apartment layout (ceiling not shown). Fig. 2b and Fig. 2c illustrate the placement of measurement points, with (b) viewed from the side and (c) from a top-down perspective.

RAM. The simulation required approximately eight days to generate the RSS measurements for all interior receivers from both interior and exterior transmitters. It is important to note that this simulation process is specific to our study solely for testing purposes; in a real-world deployment of ZeroTouch such simulations are not needed.

5.2 MATLAB

After acquiring the RSS values, we utilise MATLAB to implement the autoencoder and evaluate ZeroTouch. It enables straightforward neural network implementation using the MATLAB Deep Learning Toolbox.[§] The same computer used for Wireless InSite simulations was also used for autoencoder training.

Adding noise. The measurements obtained from Wireless InSite are static. When using the same model geometry, repeated simulations yield identical outputs, as the software does not incorporate noise. To better reflect real-world conditions, we introduced additive white Gaussian noise (AWGN) to simulate thermal noise, generating multiple samples per RSS measurement. This further improves realism, as real-world RSS measurements also fluctuate over time. In neural networks, this noise-based data augmentation enhances training by increasing data diversity and improving the model's robustness to variations and noise. We applied white noise with a signal-to-noise ratio (SNR) of 20 dB, a typical value for Wi-Fi [8].

To ensure sufficient data for reliable training, we set the noise augmentation factor to five, generating five additional noisy samples for each measurement. Each RSS-based vector ($\mathbf{x}$ vector) contains measurements from one transmitter to all anchor nodes. With noise augmentation, we effectively create five additional $\mathbf{x}$ vectors for every transmitter. For autoencoder training, these additional vectors introduce variability and enhance robustness. During evaluation, they enable independent classification of each vector, yielding separate classification results. We use a majority voting process to determine the final classification for a specific transmitter. In this process, if the majority of classification results indicate that the RSS measurements originate from an adversary device, we classify the original RSS measurements as coming from an outside transmitter.

Data normalisation. Data normalisation is a standard preprocessing step before training neural networks. We compute the mean (μ) and standard deviation (σ) of the RSS measurements from the training set and use these values to normalise the data to zero mean and unit variance. The same μ and σ are applied to the test set, ensuring consistent data transformation.

Autoencoder training. As stated in Sect. 4.1, the autoencoder is trained exclusively on RSS measurements obtained from anchor nodes. After noise-based data augmentation and normalisation, we train a simple autoencoder with a single hidden layer of size 15, using the log-sigmoid (logsig) transfer function for both the encoder and decoder. Notably, while we tested multiple autoencoder architectures, a simple design proved effective, with more complex models providing no significant improvement.

6 Evaluation of Simulation Results

In this section, we distinguish between two possible scenarios for the placement of anchor nodes within the geofence area: one without human involvement (Sect. 6.2) and the other incorporating a light *human-in-the-loop* element (Sect. 6.3). We refer to the former as the random node placement scenario and the latter as the smart node placement scenario. Both scenarios were evaluated across all three rooms within the geofence region, as defined in Sect. 5, and the results were compared to assess the security implications of each approach.

6.1 Threshold Determination

Before evaluating ZeroTouch's performance, it is essential to determine the reconstruction error threshold δ using only training (*a priori*) data. Selecting this threshold is inherently challenging, and various methods have been proposed in the literature [45]. In our case, we tested several statistical approaches based on the reconstruction errors from the training data and their statistical properties. These include thresholds calculated using the mean, median, maximum, z-score, and 95th percentile of the reconstruction errors, as well as the Interquartile Range (IQR) method and fitting Gaussian Mixture Models. Details on these methods are available in our GitHub repository. Based on our evaluations, the IQR method consistently demonstrated the best classification accuracy [46].

[§] https://www.mathworks.com/products/deep-learning.html

6.2 Scenario 1: Random placement of nodes

The performance of ZeroTouch depends on the positioning of anchor nodes within the secure geofence region. In the first scenario, we assume a random placement of the nodes, with each room in the geofence region containing *n* randomly positioned anchor nodes. This scenario represents the worst-case scenario, as we have no control over the anchor nodes' positions. We assume the anchor nodes are already deployed for other purposes, and users are not required to position them at specific locations manually. Hence, no human involvement is needed in the placement process.

As detailed in the threat model (Sect. 3.2), an attacker located outside the geofence area can increase their transmit power to raise the RSS, attempting to falsely prove their presence within the secure region. This strategy aims to offset the attenuation caused by the outer walls and align the RSS with the range expected from a genuine device. However, suppose the attacker increases the power excessively or insufficiently. In that case, the RSS measurements at the anchor nodes will fall outside the expected range, resulting in a set of RSS measurements unfamiliar to the autoencoder. This unfamiliarity leads to higher reconstruction errors, increasing the likelihood of detection. Consequently, for each outside position, the attacker is constrained to a narrow range of power increases — neither too high nor too low — to try to evade detection. We tested various power levels and found that, on average, a 7 dB increase provides the greatest advantage to the attacker, aligning with findings from the research community on signal attenuation through different materials [6]. The behaviour of different power levels and their impact on attack success will be analysed in more detail later. It is important to note that the 7 dB power increase is specific to the room and attacker configuration in our setup and should not be considered a general rule. Different apartment layouts will likely require varying optimal power increases, making it challenging for the attacker to determine the ideal value in practice.

Let us examine how the performance of ZeroTouch changes with respect to the number of randomly selected nodes. To obtain these results, we evaluated the accuracy of our model ten separate times for each specified number of nodes, recalculating the results in each trial. We then calculated the average accuracy and standard deviation across these ten evaluations. Table 1a presents the average accuracy and standard deviations for various node counts. Table 1b outlines the results for the smart node placement scenario, which will be analysed in the following section.

To provide a comprehensive view, we distinguish between two cases for each room. First, we report results for outside nodes located within half a metre of the tested room — representing the most advantageous positions for attackers due to their proximity. Second, we present results considering all 102 attacker nodes. This analysis is performed for each of the three rooms within the secure geofence area defined in Sect. 5, along with an overall average accuracy across all rooms. Based on these results, we draw the following key observations:

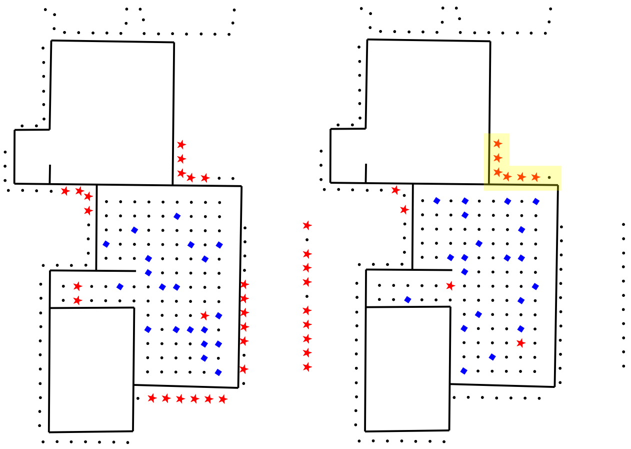
(1) **Dependence on Anchor Node Quantity**. Accuracy is influenced by the number of anchor nodes. Increasing the number of randomly placed anchor nodes generally improves accuracy, offering two main advantages. First, during the initialisation phase, having more anchor nodes provides the autoencoder with more data, enabling it to better learn the indoor RSS model of the room. Second, when a new device attempts to prove its presence inside the room, having more receivers increases the available data, making it easier to detect inconsistencies in the RSS and achieve correct classification. However, beyond a certain point, adding more nodes results in only marginal accuracy improvements. Once the room model has been sufficiently learned and the room is adequately covered with receivers, additional anchor nodes provide little to no extra benefit.

(2) **Dependence on Room Type**. Accuracy varies depending on the room type. Room one, the largest and most complex (with a non-rectangular shape and a wide variety of furniture types), has a lower initial accuracy level and requires more anchor nodes to reach the same accuracy as simpler rooms. In contrast, the smallest and simplest, room two, achieves the highest accuracy with fewer anchor nodes, as to be expected.

(3) **Proximity Node Accuracy**. The accuracy of nodes positioned close to the room is comparable to, or slightly lower than, the overall accuracy. This observation supports the claim that the most advantageous locations for an attacker are just outside the rooms.

(4) **Impact of Anchor Node Location**. The positioning of anchor nodes within a room significantly affects accuracy. The same number of anchor nodes, when positioned differently, can result in varying accuracy levels. This is expected, as not all receivers provide the same amount of information.

The average accuracy comprises the true negative rate and the true positive rate. The *true negative rate* (or true negatives) refers to ZeroTouch's ability to correctly classify genuine inside devices as inside. Conversely, if ZeroTouch incorrectly classifies an outside device as inside, this is referred to as a *false negative rate* (or false



(a) Overall accuracy of 82.4%. **(b) Overall accuracy of 94.7%.**

Figure 3: Locations of transceivers and transmitters tested within room one. Blue points ◆ represent anchor nodes, black points • indicate correct classifications, and red points ★ indicate misclassifications by ZeroTouch. Fig. 3a shows the results with a true negative rate of 96.5% and a true positive rate of 70.6%, while Fig. 3b represents the results with a true negative rate of 97.6% and a true positive rate of 92.2%.

	Room 1 [%]		**Room 2 [%]**		**Room 3 [%]**		**Average [%]**	
n	Proximity	Full	Proximity	Full	Proximity	Full	Proximity	Full
10	78.81, 3.81	77.67, 5.08	83.51, 11.28	92.50, 5.14	84.09, 4.31	86.82, 4.37	82.14	85.66
15	80.77, 4.69	79.36, 4.96	95.26, 7.26	97.02, 3.91	84.82, 4.20	87.33, 4.19	86.95	87.90
20	84.93, 7.56	83.36, 8.72	97.19, 7.08	97.99, 3.89	86.00, 5.06	87.56, 4.51	89.37	89.64
25	86.97, 6.98	84.67, 7.47	96.84, 6.50	98.46, 3.18	85.45, 5.07	87.04, 4.38	89.75	90.06
30	86.83, 6.52	84.56, 8.19	97.01, 7.02	98.66, 3.31	87.18, 3.28	88.45, 2.62	90.34	90.56
35	91.54, 4.99	89.28, 6.35	97.89, 6.65	98.97, 3.25	89.09, 0.42	89.27, 0.66	92.84	92.51
40	92.81, 6.47	91.24, 7.42	100, 0	100, 0	89.18, 0.79	89.16, 0.86	93.99	93.46

(a) Average accuracy with random node placement.

	Room 1 [%]		**Room 2 [%]**		**Room 3 [%]**		**Average [%]**	
n	Proximity	Full	Proximity	Full	Proximity	Full	Proximity	Full
10	82.18, 5.51	81.95, 6.73	92.11, 9.18	96.23, 4.48	86.27, 3.36	88.94, 3.07	86.85	89.04
15	88.38, 5.80	88.49, 6.51	97.72, 3.10	98.89, 1.39	88.27, 2.12	89.60, 2.03	91.46	92.33
20	91.41, 6.24	91.23, 6.88	99.47, 1.66	99.75, 0.79	89.18, 1.45	90.50, 1.12	93.35	93.83
25	94.22, 4.08	93.20, 5.41	N/A	N/A	89.45, 0.98	90.49, 0.87	94.38	94.48
30	94.86, 4.48	93.28, 4.98	N/A	N/A	89.27, 0.58	90.08, 0.67	94.53	94.37
35	96.69, 0.67	95.66, 1.11	N/A	N/A	N/A	N/A	95.14	95.16
40	97.11, 0.84	96.22, 1.06	N/A	N/A	N/A	N/A	95.28	95.35

(b) Average accuracy with smart node placement.

Table 1: Overall classification accuracy of ZeroTouch for varying numbers of anchor nodes under two placement strategies: random node placement (Table 1a) and smart node placement (Table 1b). Each table shows the average accuracy and standard deviation, covering two attacker cases: attacker nodes positioned in close proximity to each room (Proximity) and all attacker positions around the rooms (Full).

negatives). Similarly, the *true positive rate* (or true positives) refers to ZeroTouch's ability to correctly classify outside devices as outside. Finally, when ZeroTouch incorrectly classifies a genuine inside device as outside, this is referred to as the *false positive rate* (or false positives).

Determining the true positive and true negative rates requires analysing which nodes were correctly and incorrectly classified, as well as their respective locations relative to the geofence area. Given that room one is the largest and most complex room, we use it as an example to illustrate the classification results, as shown in Fig. 3. Additional visualisations of the results can be found in Appx. B or in our GitHub repository. The figure, generated in MATLAB, presents two different anchor node configurations with the same number of anchor nodes (20). In both configurations, ZeroTouch achieves high accuracy in the true negative rate, whereas the true positive rate exhibits greater variability. Our testing indicates that this pattern generally holds consistently across all tested rooms, not just room one. Furthermore, as noted previously, the positioning of anchor nodes significantly impacts the overall accuracy. In Fig. 3b, both the true positive and true negative rates are higher than in Fig. 3a despite the same number of anchor nodes used in both scenarios.

Although RSS is inherently unpredictable, we can still analyse the results to understand misclassifications. As shown in Fig. 3, certain outside nodes are consistently misclassified. Notably, the yellow-highlighted corner in Fig. 3b is persistently misclassified, proving to be the most challenging area for room one. Across multiple experiments, even with more anchor nodes, outside transmitters in this corner are often classified as being inside room one. Referring to the apartment layout in Fig. 2a, we see this corner contains a wooden door, which attenuates signals less than walls. Additionally, a wall north of the transmitters introduces reflections due to their omnidirectional antennas. These reflections propagate into the room, resembling the signal behaviour of an actual device near an interior door. This makes it harder for ZeroTouch to distinguish these outside transmitters from genuine in-room devices.

We can also observe this phenomenon in Fig. 4, which represents a stem plot of residual errors per transmitter for a test set, along with the threshold used for classification. The plot corresponds to the classification shown in Fig. 3b. Two distinct regions can be observed: the red-shaded region, which corresponds to the outside transmitters (102 in total), and the green-striped region, which corresponds to the inside transmitters. By comparing the residual errors to the threshold, we note that eight outside transmitters produce residual errors below the threshold. These are incorrectly classified as inside devices, resulting in false negatives (cf. the locations of these transmitters in Fig. 3b). Notably, the residual error of the yellow corner is comparable to those of the inside transmitters, explaining the difficulty ZeroTouch faces in classifying these transmitters correctly. Conversely, for transmitters inside the room, we observe that two produce residual errors above the threshold. These are incorrectly classified as outside devices, resulting in false positives. (cf. the locations of these transmitters in Fig. 3b). The residual errors plot in Fig. 4 also underscores the importance of selecting an appropriate threshold value. A higher threshold leads to more false negatives, while a lower threshold increases false positives, making it crucial to strike a balance between the two.

Per-Position Power Sweep. A 7 dB increase represents the most advantageous *average* case for the attacker. To further explore the attacker's potential advantage, we extended our analysis by testing

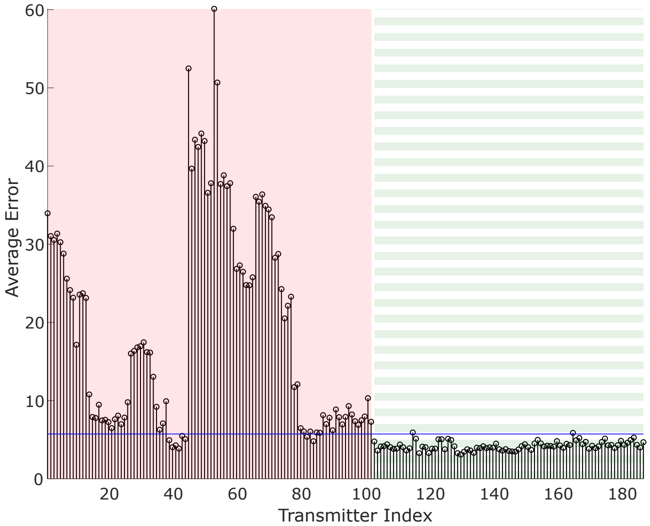


Figure 4: Stem plot illustrating the average errors of the test set for each transmitter. The red-shaded region represents transmitters located outside the room. The green-striped region represents transmitters located inside the room. The horizontal blue line represents the classification threshold.

various power levels at each outside attacker position. Specifically, we conducted a power sweep from 0 dB to 100 dB in 0.25 dB increments for each outside location to determine the power increase that results in the lowest reconstruction error — i.e., the optimal attack scenario for that position. As a result, each outside location is assigned a specific, non-uniform power increase. It is important to note that this analysis is theoretical and represents an optimal case for the attacker. In practice, the attacker does not have access to the reconstruction error values from the autoencoder and can only observe whether their attack succeeds or fails. In a real-world scenario, the attacker would need to iteratively test different transmit power levels, adjusting to maximize their chances of success — but without any guarantee of achieving it. For this analysis, we used the same randomly selected anchor node locations as in Fig. 3b.

To understand the impact of the power sweep on accuracy, we refer to Fig. 5, which compares the false positive and false negative rates between the 7 dB increase and the power sweep attack strategy as a function of the threshold value. We observe that the false positive rates remain identical for both cases. This is expected, as the threshold is determined a priori from anchor node measurements, and indoor genuine devices are unaffected by different attacker strategies. However, when comparing the false negative curves, we notice a leftward shift in the power sweep case. As a result, for the same threshold value, false negatives are higher compared to the 7 dB scenario. This also aligns with expectations, as more outside transmitters can now deceive ZeroTouch and be incorrectly classified as inside. From our testing, we observed an approximately 10% drop in accuracy compared to the 7 dB case. Since this analysis represents a theoretically optimal attack case, in practice, the accuracy drop will be smaller. Nonetheless, these findings highlight the effectiveness of aggregating RSS measurements across multiple receivers to enhance security, as many outside locations remain where the attacker cannot succeed, regardless of the chosen transmit power.

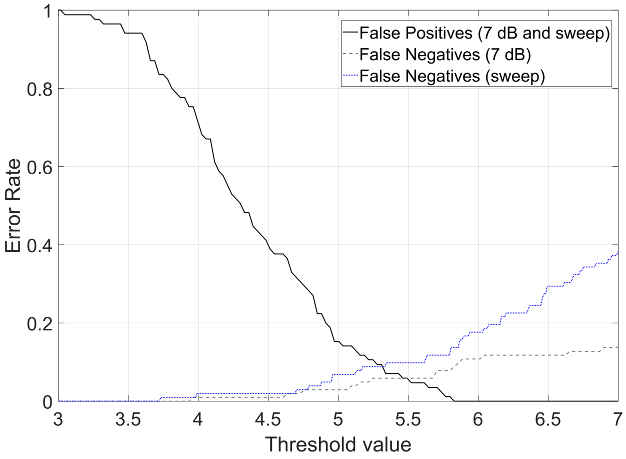
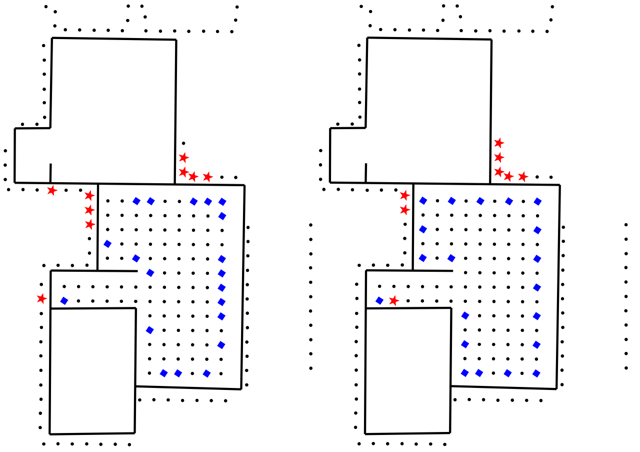


Figure 5: False positive and false negative rates as a function of the threshold value for the 7 dB fixed power increase and power sweep attack strategies.



(a) Overall accuracy of 95.2%. **(b) Overall accuracy of 95.7%.**

Figure 6: Fig. 6a shows the results with a true negative rate of 100% and a true positive rate of 91.2%, while Fig. 6b represents the results with a true negative rate of 98.8% and a true positive rate of 93.1%.

6.3 Scenario 2: Smart placement of nodes

We have observed that the placement of anchor nodes significantly influences the quality of RSS measurements and their contribution to classification accuracy, as not all locations are equally effective. Identifying anchor node positions that provide the most informative measurements can help enhance ZeroTouch's accuracy. Our tests found that anchor nodes positioned closer to the inner walls provide more useful RSS data, leading to better classification performance. We refer to this specific selection of inner-wall anchor nodes as a smart placement of nodes. The smart node placement may require a light *human-in-the-loop* element, with basic guidelines recommending that anchor nodes be placed as close as possible to the inner walls of a room. It is important to note that this may not always be straightforward for certain fixed devices, such as smart smoke detectors. However, in many cases, smart devices are already positioned near walls (e.g., smart plugs, thermostats, light switches, sensors), suggesting that the placement of anchor nodes is not entirely random, as assumed in Scenario 1. Additionally, some devices, such as smart speakers, can be easily relocated to more favourable positions. Therefore, achieving or approximating a smart anchor node configuration should be feasible in most real-world scenarios. The results for various anchor node counts, where nodes are selected randomly but exclusively from the set of inner-wall anchor nodes specific to a given room, are summarised in Table 1b. The table presents the classification accuracy when the attacker applies a uniform power increase of 7 dB. The cells marked as N/A indicate scenarios where there are insufficient inner-wall anchor nodes available in the room to meet the required node count. Compared to the results in Table 1a, we observe that smart placement achieves higher accuracy for the same number of anchor nodes. For instance, twenty randomly selected anchor nodes from the smart placement set yield, on average, higher accuracy (93.83%) than twenty completely random anchor nodes (89.64%). This also demonstrates that smart placement requires fewer nodes to achieve similar accuracy levels. In Fig. 6, we visualise the results for room one, chosen again as the largest and most complex room. The figure presents two different anchor node configurations, each using the same number of anchor nodes as before (20) and using the smart placement strategy. In Fig. 6a, the twenty anchor nodes were randomly selected from the inner-wall set, while in Fig. 6b, every second anchor node from the inner-wall set was selected. The results show that both selection strategies yield comparable accuracy. Additionally, both approaches achieve better accuracy compared to the scenario where anchor nodes are selected completely at random.

7 Discussion

In this section, we evaluate the strengths and limitations of ZeroTouch, highlighting its scalability, flexibility, and practical applicability in real-world scenarios. We demonstrate how ZeroTouch serves as a robust, *plug-and-play* solution that minimises the *human-in-the-loop* element while addressing critical challenges in secure geofencing and IoT device commissioning.

Scalability. Scalability is one of the key advantages of ZeroTouch. By scalability, we mainly mean that the execution time increases linearly with the number of devices being verified. This is because each device verification is inherently an independent event. Additionally, the verification process is highly efficient, requiring only the collection of RSS measurements and running the pre-trained autoencoder. As a result, ZeroTouch allows users to verify large numbers of devices with ease.

Flexibility. Depending on their needs, users have the flexibility to implement ZeroTouch in different ways. If they prefer a fully automated system without any *human-in-the-loop* element, ZeroTouch can be deployed using existing devices already present in the room. This corresponds in worst case to Scenario 1, as described in Sect 6.2. Alternatively, if users value the improved accuracy of Scenario 2, they may opt for a light human-in-the-loop approach, where anchor nodes are positioned more strategically, as outlined in Sect. 6.3. This flexibility allows users to adapt the system to their priorities and requirements.

Since classification results depend on the chosen threshold, users can tailor the system to prioritise either security or usability. Adjusting the threshold affects the balance between the true positive rate (associated with security) and the true negative rate (associated with usability). Our method aims to strike a balanced trade-off between these two objectives. In this work, we have presented results that reflect such a balanced configuration. However, the threshold can be adapted to meet specific operational goals. For instance, increasing it to favour usability (resulting in a higher true negative rate and a lower true positive rate), or decreasing it to favour security (yielding a higher true positive rate and lower true negatives). This flexibility allows for custom thresholding strategies aligned with specific priorities.

Use cases. The potential applications of ZeroTouch span various IoT deployment scenarios, particularly those where secure geofencing is essential but traditionally difficult to achieve autonomously. Conventional device commissioning often involves manual configuration, such as scanning QR codes [9], gesture-based pairing [20, 22], or manually assigning devices to zones [12], all of which are inefficient and prone to human error. ZeroTouch addresses these challenges by providing an autonomous and frictionless security enhancement to existing authentication methods. Instead of replacing existing authentication procedures, ZeroTouch complements them by passively verifying that devices claiming to be within a secure area genuinely are. When a device attempts to join the network, ZeroTouch automatically analyses the RSS signatures from multiple anchor points, ensuring accurate, zero-interaction location verification. If the device is confirmed within the designated boundary, it is seamlessly registered. Otherwise, the device is flagged as potentially malicious, and network access is denied. This automated process significantly reduces human involvement, enhancing usability and overall security without additional deployment complexity.

Beyond commissioning, ZeroTouch has direct implications for access control in high-security environments and healthcare systems, where enforcing strict physical boundaries is critical to operational safety [7]. ZeroTouch can be used as an additional security layer to verify that only authorised IoT devices physically present within the premises are granted network access. For example, any device attempting to connect from outside the building or restricted zone would be immediately identified as unauthorised, ensuring that adversarial attempts to spoof locations or gain remote access are denied. Similarly, in healthcare settings, ZeroTouch can enhance operational security by ensuring that medical devices remain within their assigned rooms [1]. For instance, critical devices such as infusion pumps or patient monitors must operate only within designated hospital rooms and medical devices are automatically commissioned and assigned to their respective rooms. ZeroTouch is also a promising solution for IoT home automation systems, where it can enforce room-specific rules by ensuring that devices such as voice assistants, smart speakers, or cameras operate only within designated areas.

8 Conclusion

In this paper, we presented *ZeroTouch*, a system for secure geofencing leveraging Received Signal Strength (RSS) measurements and machine learning. ZeroTouch addresses key challenges in IoT device localisation by providing an automated, practical mechanism to verify device locations within defined boundaries. By aggregating inherently available RSS data from multiple receivers, our approach effectively detects anomalies in location claims, thus offering a layered security enhancement to existing authentication methods. Our work demonstrates that RSS remains relevant in practical security applications, despite its intrinsic theoretical limitations. Since RSS measurements are natively supported by most IoT devices, ZeroTouch provides an essentially cost-free security upgrade without additional hardware or manual calibration.

We validated ZeroTouch in a realistic simulation environment, demonstrating its ability to achieve over 90% accuracy under optimal conditions. This highlights its potential for real-world applications, including secure IoT device commissioning, access control, and smart home automation.

Acknowledgments

This work was supported in part by CyberSecurity Research Flanders with reference number VR20192203, and by the European Union's Horizon Research and Innovation program under grant agreement No. 101119747 (TELEMETRY).

References

[1] AirDroid Team. 2023. Geofencing in Healthcare: Benefits and Applications. https://blog.airdroid.com/post/geofencing-in-healthcare/.

[2] S. Abhishek Anand and Nitesh Saxena. 2016. Vibreaker: Securing Vibrational Pairing with Deliberate Acoustic Noise. In *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks* (Darmstadt, Germany) *(WiSec '16)*. 103–108.

[3] Chris Bonebrake and Lori Ross O'Neil. 2014. Attacks on GPS Time Reliability. *IEEE Security & Privacy* 12, 3 (2014).

[4] Srdjan Capkun, Saurabh Ganeriwal, Farooq Anjum, and Mani B. Srivastava. 2011. Secure RSS-based localization in sensor networks.

[5] Raghavendra Chalapathy and Sanjay Chawla. 2019. Deep Learning for Anomaly Detection: A Survey. *arXiv preprint arXiv:1901.03407* (2019).

[6] Common, Losses Through. 2002. *Propagation Losses Through Common Building Materials: 2.4 GHz vs 5 GHz.* E10589.

[7] Forbes Technology Council. 2024. How Geofencing Tech Can Boost Private And Public Sector Processes. https://www.forbes.com/councils/forbestechcouncil/2024/10/23/how-geofencing-tech-can-boost-privateand-public-sector-processes/.

[8] Meraki Documentation. 2024. Signal-to-Noise Ratio (SNR) and Wireless Signal Strength. https://documentation.meraki.com/MR/Wi-Fi_Basics_and_Best_Practices/Signal-to-Noise_Ratio_%28SNR%29_and_Wireless_Signal_Strength Accessed: 2024-11-18.

[9] Sayon Duttagupta, Eduard Marin, Dave Singelée, and Bart Preneel. 2023. HAT: Secure and Practical Key Establishment for Implantable Medical Devices. In *Proceedings of the Thirteenth ACM Conference on Data and Application Security and Privacy (CODASPY '23)*. 213–224.

[10] Ravi Garg, Avinash L. Varna, and Min Wu. 2012. An Efficient Gradient Descent Approach to Secure Localization in Resource Constrained Wireless Sensor Networks. *IEEE Transactions on Information Forensics and Security* (2012).

[11] Mohammad Reza Gholami, Reza Monir Vaghefi, and Erik G. Ström. 2013. RSS-Based Sensor Localization in the Presence of Unknown Channel Parameters. *IEEE Transactions on Signal Processing* (2013).

[12] Nirnimesh Ghose, Loukas Lazos, and Ming Li. 2018. SFIRE: Secret-Free-in-band Trust Establishment for COTS Wireless Devices. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications*. 1529–1537.

[13] Kai Jansen, Matthias Schäfer, Daniel Moser, Vincent Lenders, Christina Pöpper, and Jens Schmitt. 2018. Crowd-GPS-Sec: Leveraging Crowdsourcing to Detect and Localize GPS Spoofing Attacks. In *2018 IEEE Symposium on Security and Privacy (SP)*. 1018–1031.

[14] Sunawar Khan, Tehseen Mazhar, Tariq Shahzad, Afsha Bibi, Wasim Ahmad, Muhammad Amir Khan, Mamoon M Saeed, and Habib Hamam. 2024. Antenna systems for IoT applications: a review. *Discover Sustainability* 5, 1 (2024).

[15] Justin Larouzee and Jean-Christophe Le Coze. 2020. Good and bad reasons: The Swiss cheese model and its critics. *Safety Science* 126 (2020).

[16] Loukas Lazos and Radha Poovendran. 2004. SeRLoc: secure range-independent localization for wireless sensor networks. In *Proceedings of the 3rd ACM Workshop on Wireless Security (WiSe '04)*.

[17] Chaohao Li, Xiaoyu Ji, Bin Wang, Kai Wang, and Wenyuan Xu. 2021. SenCS: Enabling Real-time Indoor Proximity Verification via Contextual Similarity. *ACM Trans. Sen. Netw.* 17, 2, Article 19 (may 2021), 22 pages.

[18] Pengzhi Li, Yan Pei, and Jianqiang Li. 2023. A comprehensive survey on design and application of autoencoder in deep learning. *Applied Soft Computing* 138 (2023), 110176. doi:10.1016/j.asoc.2023.110176

[19] Xinrong Li. 2006. RSS-Based Location Estimation with Unknown Pathloss Model. *IEEE Transactions on Wireless Communications* (2006).

[20] Xiaopeng Li, Qiang Zeng, Lannan Luo, and Tongbo Luo. 2020. T2Pair: Secure and Usable Pairing for Heterogeneous IoT Devices. In *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security (CCS '20)*.

[21] Claudio Marforio, Nikolaos Karapanos, Claudio Soriente, Kari Kostiainen, and Srdjan Capkun. 2014. Smartphones as Practical and Secure Location Verification Tokens for Payments. In *Proceedings of the Network and Distributed System Security Symposium Symposium, NDSS '14*.

[22] Rene Mayrhofer and Hans Gellersen. 2009. Shake Well Before Use: Intuitive and Secure Pairing of Mobile Devices. *IEEE Transactions on Mobile Computing* 8, 6 (2009), 792–806.

[23] Jonathan M. McCune, Adrian Perrig, and Michael K. Reiter. 2005. Seeing-is-believing: using camera phones for human-verifiable authentication. In *2005 IEEE Symposium on Security and Privacy (S P'05)*. 110–124.

[24] Petar Medeđović, Mladen Veletić, and Željko Blagojević. 2012. Wireless insite software verification via analysis and comparison of simulation and measurement results. In *2012 Proceedings of the 35th International Convention MIPRO*. 776–781.

[25] Qi Mi, John A. Stankovic, and Radu Stoleru. 2010. Secure walking GPS: a secure localization and key distribution scheme for wireless sensor networks. In *Proceedings of the Third ACM Conference on Wireless Network Security (WiSec '10)*.

[26] Maryam Motallebighomi, Harshad Sathaye, Mridula Singh, and Aanjhan Ranganathan. 2023. Location-independent GNSS Relay Attacks: A Lazy Attacker's Guide to Bypassing Navigation Message Authentication. In *Proceedings of the 16th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '23)*. 365–376.

[27] Bodhibrata Mukhopadhyay, Seshan Srirangarajan, and Subrat Kar. 2018. Robust Range-Based Secure Localization in Wireless Sensor Networks. In *2018 IEEE Global Communications Conference (GLOBECOM)*.

[28] Bodhibrata Mukhopadhyay, Seshan Srirangarajan, and Subrat Kar. 2021. RSS-Based Localization in the Presence of Malicious Nodes in Sensor Networks. *IEEE Transactions on Instrumentation and Measurement* (2021).

[29] Ben Nassi, Ron Bitton, Ryusuke Masuoka, Asaf Shabtai, and Yuval Elovici. 2021. SoK: Security and Privacy in the Age of Commercial Drones. In *2021 IEEE Symposium on Security and Privacy (SP)*. 1434–1451.

[30] Cam Ly Nguyen and Aftab Khan. 2017. WiLAD: Wireless Localisation through Anomaly Detection. In *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*.

[31] Konstantin Ninković. 2024. CAD Model of an Apartment. Private communication. https://www.zdl.studio/en Created while employed at ZDL Studio.

[32] Huthaifa A Obeidat, Omar A Obeidat, Mahmood F Mosleh, Ali A Abdullah, and Raed A Abd-Alhameed. 2020. Verifying received power predictions of wireless insite software in indoor environments at WLAN frequencies. *The Applied Computational Electromagnetics Society Journal (ACES)* (2020), 1119–1126.

[33] Avinash Nanasaheb Pawar and Shankar B Deosarkar. 2024. A comprehensive review of different antennas for IoT applications. *Wireless Networks* (2024), 1–13.

[34] Marcin Poturalski, Manuel Flury, Panos Papadimitratos, Jean-Pierre Hubaux, and Jean-Yves Le Boudec. 2012. On Secure and Precise IR-UWB Ranging. *IEEE Transactions on Wireless Communications* (2012).

[35] Remcom Inc. 2022. *Wireless InSite, Version 3.4.4*. http://www.remcom.com/wireless-insite

[36] Sandro Rodriguez Garzon and Bersant Deva. 2014. Geofencing 2.0: taking location-based notifications to the next level. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp '14)*.

[37] Salvatore Scellato, Mirco Musolesi, Cecilia Mascolo, Vito Latora, and Andrew T. Campbell. 2011. NextPlace: A Spatio-Temporal Prediction Framework for Pervasive Systems. In *Proceedings of the 9th International Conference on Pervasive Computing* (San Francisco, USA) *(Pervasive'11)*. 152–169.

[38] Domien Schepers, Mridula Singh, and Aanjhan Ranganathan. 2021. Here, there, and everywhere: security analysis of wi-fi fine timing measurement. In *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '21)*.

[39] The MathWorks, Inc. 2023. *MATLAB and Simulink R2023b*. https://www.mathworks.com/products/matlab.html

[40] Xiaohua Tian, Ruofei Shen, Duowen Liu, Yutian Wen, and Xinbing Wang. 2017. Performance Analysis of RSS Fingerprinting Based Indoor Localization. *IEEE Transactions on Mobile Computing* 16, 10 (2017), 2847–2861. doi:10.1109/TMC.2016.2645221

[41] Nils Ole Tippenhauer and Srdjan Čapkun. 2009. ID-Based Secure Distance Bounding and Localization. In *Computer Security – ESORICS 2009*.

[42] Nils Ole Tippenhauer and Srdjan Capkun. 2012. UWB-based secure ranging and localization. *Technical Report – ETH Zurich, Department of Computer Science* 586 (2012).

[43] Nils Ole Tippenhauer, Christina Pöpper, Kasper Bonne Rasmussen, and Srdjan Capkun. 2011. On the requirements for successful GPS spoofing attacks. In *Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS '11)*. 75–86.

[44] Slavisa Tomic, Marko Beko, and Rui Dinis. 2015. RSS-Based Localization in Wireless Sensor Networks Using Convex Relaxation: Noncooperative and Cooperative Schemes. *IEEE Transactions on Vehicular Technology* (2015).

[45] Hasan Torabi, Seyedeh Leili Mirtaheri, and Sergio Greco. 2023. Practical autoencoder based anomaly detection by using vector reconstruction error. *Cybersecurity* 6, 1 (2023), 1. doi:10.1186/s42400-022-00134-9

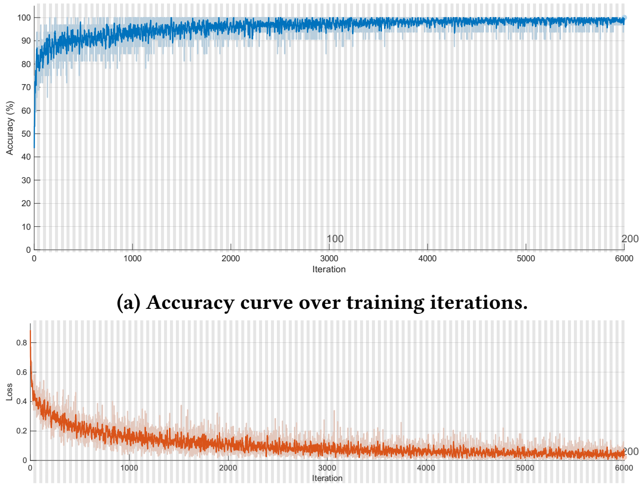
[46] HP Vinutha, B Poornima, and BM Sagar. 2018. Detection of outliers using interquartile range technique from intrusion dataset. In *Information and decision sciences: Proceedings of the 6th international conference on ficta*. Springer, 511–518.

[47] Wi-Fi Alliance. 2024. Wi-Fi Aware | Wi-Fi Alliance. https://www.wi-fi.org/discover-wi-fi/wi-fi-aware.

[48] Jie Yang and Yingying Chen. 2009. Indoor Localization Using Improved RSS-Based Lateration Methods. In *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*.

[49] Lizhou Yuan, Yidan Hu, Yunzhi Li, Rui Zhang, Yanchao Zhang, and Terri Hedgpeth. 2018. Secure RSS-Fingerprint-Based Indoor Positioning: Attacks and Countermeasures. In *2018 IEEE Conference on Communications and Network Security (CNS)*.

[50] Jiansong Zhang, Zeyu Wang, Zhice Yang, and Qian Zhang. 2017. Proximity based IoT device authentication. In *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*. 1–9.



(a) Accuracy curve over training iterations.

(b) Loss curve over training iterations.

Figure 7: Accuracy and loss during the learning process of the supervised learning baseline.

A Supervised Learning Baseline

In our case, the supervised learning baseline is implemented as a simple multilayer perceptron (MLP) neural network. Similar to the autoencoder, it takes $\mathbf{x}$, an RSS-based vector representing the RSS values recorded by N anchor nodes as input. The network consists of a single fully connected hidden layer with a size of 15 neurons (matching the size of the autoencoder's hidden layer) and uses a Rectified Linear Unit (ReLU) activation function. The output layer is a softmax layer with two neurons corresponding to the two classes: 'outside' and 'inside.' The Adam optimiser is used for training, with an initial learning rate of 0.001. The validation frequency is set to 50, and the mini-batch size is set to 32. Further details of the network are available on our GitHub repository.

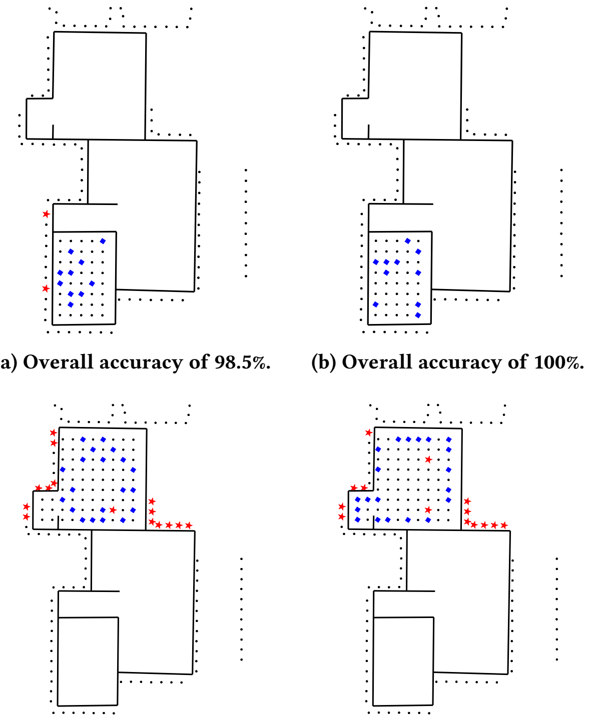
Since this is a supervised learning approach, the RSS-based vectors ($\mathbf{x}$) are labelled to indicate whether they originate from an inside or outside transmitter. As with the autoencoder, noise is added to the measurements for augmentation, the data is normalised, and majority voting is used to determine the final classification. The goal here is to evaluate the network's accuracy in an ideal environment by training it on all transmitter locations (using the full set of $\mathbf{x}$ vectors) and then evaluating its performance on the same data. Although this approach is not typical in practice (using the same set for training and evaluation), it allows us to assess whether the neural network has sufficient capacity to fit the training data. Additionally, it provides insight into the theoretical maximum accuracy achievable with the given set of anchor nodes.

While the network is expected to overfit the data, the small size of the hidden layer (15 neurons) makes overfitting more challenging. The training process generally takes longer compared to the autoencoder, ranging from 1 to 5 minutes. An example of the training procedure where twenty anchor nodes are used is shown in Fig. 7, where the convergence of the accuracy and loss curves can be observed. In this example, a very good accuracy of over 95% is achieved.

We compare the results of the autoencoder and the supervised learning approach in scenarios where both networks use the same anchor nodes for classification. From our testing, we observe that the autoencoder performs worse than supervised learning, as expected. For smaller numbers of anchor nodes (up to 20), the performance gap is relatively small — within 10%. This indicates that the autoencoder achieves results that are quite close to the theoretical maximum achievable with that set of anchor nodes. However, with a greater number of anchor nodes, the difference in performance becomes more pronounced. The supervised learning baseline consistently achieves very high accuracy, almost always exceeding 95%, while the autoencoder exhibits more variability in its results. For instance, in certain cases with less optimal ('bad') anchor node placements, the autoencoder achieves accuracies as low as 80%. This demonstrates that supervised learning is better able to leverage a larger number of receivers to improve classification performance.

B Supplementary Figures

This appendix provides additional visualisations of the results (see Fig. 3), including further tested transceivers and transmitters, particularly for rooms two and three in the apartment.



(a) Overall accuracy of 98.5%. **(b) Overall accuracy of 100%.**

(c) Overall accuracy of 90.6%. **(d) Overall accuracy of 91.2%.**

Figure 8: Fig. 8a shows the results with a true negative rate of 100% and a true positive rate of 98%, while Fig. 8b represents the results with a true negative rate of 100% and a true positive rate of 100%. Fig. 8c shows the results with a true negative rate of 98.3% and a true positive rate of 86.3%, while Fig. 8d represents the results with a true negative rate of 96.6% and a true positive rate of 88.2%.