CHALMERS

Identification of Wi-Fi-Enabled Drones via Software-Defined Radio (SDR) and Artificial Intelligence (AI)

Bachelor’s thesis in Electrical Engineering

Sam Ghasemi

Department of Microtechnology and Nanoscience - MC2
Chalmers University of Technology
Gothenburg, Sweden 2025

Abstract

This thesis presents a method for detecting Wi-Fi-controlled drones using software-defined radio (SDR) technology combined with artificial intelligence (AI). Radio frequency (RF) signals in the 2.4 GHz band were captured and analyzed to distinguish drone transmissions from conventional wireless activity. A rule-based bandwidth analysis was first implemented, followed by a convolutional neural network (CNN) classifier trained on power spectral density (PSD) features. The system successfully identified drone signals in real time under test conditions, demonstrating that SDR combined with AI provides a cost-effective and extensible framework for RF-based drone detection.

Sammanfattning

Denna rapport presenterar en metod för att upptäcka Wi-Fi-styrda drönare med hjälp av mjukvarudefinierad radio (SDR) och artificiell intelligens (AI). Radiosignaler i 2,4 GHz-bandet samlades in och analyserades för att skilja drönarkommunikation från annan trådlös trafik. Först utvecklades en regelbaserad metod för bandbreddsanalys, som därefter kompletterades med en konvolutionell neuronnätsmodell (CNN) tränad på spektrala densitetsdata (PSD). Systemet kunde identifiera drönarsignaler i realtid under testförhållanden och resultaten visar att kombinationen av SDR och AI kan erbjuda en kostnadseffektiv och flexibel lösning för RF-baserad drönardetektering.

Table of Contents

1. Introduction

1.1 Background

In recent years, the use of unmanned aerial vehicles (UAVs), commonly referred to as drones, has expanded rapidly across a wide range of sectors. Applications include agriculture, logistics, photography, surveillance, and emergency response. Drones have proven to be powerful tools for improving efficiency and providing access to environments that are otherwise difficult or hazardous. Their affordability, portability, and ease of use have also contributed to their growing popularity among both professionals and hobbyists.

At the same time, this widespread adoption has introduced significant challenges. The same characteristics that make drones useful, such as their mobility, remote controllability, and compact design, also create security risks. Unauthorized drone activity has been reported near airports, critical infrastructure, and restricted airspace. In addition, drones have been used for smuggling, espionage, and even as improvised weapons in conflict zones. These risks have increased the demand for reliable and affordable methods to detect and classify drone activity.

One category that has become especially common is Wi-Fi-based consumer drones. These drones communicate with a controller, typically a smartphone or tablet, using standard 2.4 GHz Wi-Fi protocols. They are widely available, inexpensive, and easy to operate, which makes them attractive to recreational users but also to individuals with malicious intent.

Although such drones are small and often perceived as harmless, their use in urban areas presents unique challenges. Because they rely on the same Wi-Fi protocols as everyday devices, their signals blend into ordinary network activity and are therefore difficult to detect using traditional tools. Despite their limited range, they can still be used to invade privacy, bypass barriers, or enter restricted zones without raising immediate suspicion. For these reasons, even short-range Wi-Fi-controlled drones require active monitoring in sensitive environments where unauthorized aerial activity must be quickly identified and addressed.

1.2 Objective

This project is guided by three primary objectives:

1. **Real-time Drone Detection**: To develop a system capable of detecting Wi-Fi-enabled drones within their maximum control range in real time.

2. **RF Signal Classification**: To distinguish drone transmissions from other wireless activity in the 2.4 GHz band, including Wi-Fi, Bluetooth, and related signals.

3. **Affordability and Accessibility**: To provide a cost-effective and practical alternative to professional systems, ensuring that drone detection can be implemented without the need for specialized or high-end equipment.

1.3 Scope and Limitations

This project is limited to the detection of short-range Wi-Fi-enabled consumer drones operating in the 2.4 GHz frequency band. All testing and validation were performed using a single drone model, the *DJI Tello*, as it was the only device available throughout the development process. The system has therefore been optimized to recognize the specific RF emission patterns of this model. While the methodology is intended to generalize to other Wi-Fi-based drones, reliable detection of different makes and models may require further adaptation or additional training.

The project does not include spatial localization or tracking of drones, nor does it address non-Wi-Fi-controlled drones, such as those using proprietary RF links, the 5.8 GHz band, or GPS-only systems. Furthermore, the system is not designed to block or interfere with drone signals and functions solely as a passive detection tool.

In line with the project objectives, only low-cost hardware has been used. The setup consists of an SDR with a compact antenna connected to a laptop, which ensures affordability and accessibility but imposes limitations on overall performance. Under the test conditions, the effective detection capability was limited by the practical control constraints of the DJI Tello. This will be discussed further in chapter 6.

2. Theoretical Framework

Wireless communication systems rely on a combination of physical and computational principles to transmit, capture, and interpret signals across the electromagnetic spectrum. This chapter outlines the key concepts necessary to understand signal behavior in the 2.4 GHz frequency band and the analytical tools used in this project. Core areas include the structure of the 2.4 GHz band, common signal types present within it, and the principles behind *Software Defined Radio (SDR)* technology. The chapter also describes essential signal processing methods such as the *Fast Fourier Transform (FFT)* and *bandwidth* estimation, followed by an overview of *machine learning* techniques relevant to radio signal classification.

2.1 The 2.4 GHz Frequency band

The 2.4 GHz frequency band is part of the Industrial, Scientific, and Medical (ISM) spectrum, which is globally available for unlicensed use. It spans from 2.400 GHz to 2.4835 GHz and is widely utilized due to its accessibility and compatibility with numerous communication standards, including IEEE 802.11b/g/n (Wi-Fi), Bluetooth, Zigbee, and other proprietary wireless protocols [1][2].

A notable characteristic of the 2.4 GHz band is its global uniformity. Unlike many other frequency bands that vary in allocation by country, the 2.4 GHz ISM band is consistent across most regions, making it highly attractive for consumer electronics. This universality has contributed to the widespread adoption of Wi-Fi and related technologies. However, the ubiquity of the band also results in congestion, as multiple devices compete for spectrum access, which in turn leads to interference [3].

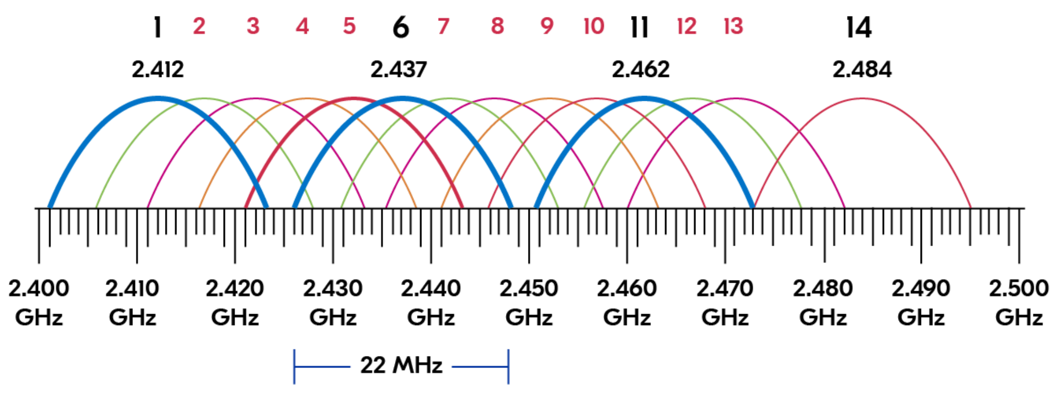

Fig. 1. Illustration of the 2.4 GHz Wi-Fi band showing the 14 available channels, their center frequencies, and channel overlap [22].

The 2.4 GHz ISM band is divided into 14 channels, each spaced 5 MHz apart. In most countries, only the first 11 channels (2.412 GHz to 2.462 GHz) are permitted for Wi-Fi use. Each channel serves as a center frequency around which a wireless device can transmit and receive data. When a device initiates a transmission, it selects one of these channels either manually or automatically, depending on availability and interference levels. Although the channels are separated by 5 MHz, typical Wi-Fi transmissions occupy a much wider bandwidth. Standard Wi-Fi signals often use a channel width of approximately 22 MHz, which extends well beyond the channel's center frequency and overlaps with adjacent channels [4].

Because of this overlap, only three channels in the 2.4 GHz band—channels 1, 6, and 11—are spaced far enough apart to avoid interference. These non-overlapping channels are particularly important in environments with many wireless devices, as they reduce mutual interference and improve communication stability. For this reason, most Wi-Fi routers are configured, either automatically or by default, to transmit on channels 1, 6, or 11. This practice minimizes overlap with neighboring networks and increases reliability, especially in densely populated areas [4].

Channel 14, centered at 2.484 GHz, differs from the others because it is spaced 12 MHz above channel 13 and is only permitted in select regions such as Japan. It is also limited to older Wi-Fi standards like IEEE 802.11b, which makes it largely irrelevant for modern communication systems. Similarly, channels 12 and 13, while permitted in some regions such as Europe, are not universally supported. In countries like the United States, they are restricted or disabled by default. As a result, many consumer devices and routers are configured for global compatibility and avoid using these channels to reduce connectivity issues [1][2].

2.2 Signal types (2.4 GHz)

The 2.4 GHz ISM band supports a wide range of wireless technologies, but two signal types dominate in both prevalence and application: *Wi-Fi* and *Bluetooth*. These protocols serve different purposes and rely on distinct modulation and access methods, yet they frequently coexist in the same physical environment, which often results in interference. Understanding their spectral and modulation characteristics is essential for analyzing activity within the band. For the purposes of drone detection, this knowledge is particularly important because Wi-Fi-based drones operate in the same spectrum as everyday wireless devices, making reliable classification a key challenge.

2.2.1 Wi-Fi (IEEE 802.11b/g/n)

Wi-Fi is the dominant wireless communication protocol in the 2.4 GHz ISM band. It is defined by the *IEEE 802.11* family of standards, with *802.11b, 802.11g*, and *802.11n* being the most widely implemented within this frequency range [4].

The first of these, *802.11b*, was introduced in 1999. It employs *Direct Sequence Spread Spectrum (DSSS)* modulation and supports data rates up to 11 Mbps. Later versions, such as *802.11g* and *802.11n*, adopted *Orthogonal Frequency Division Multiplexing (OFDM)*. This shift enabled higher throughput, improved spectral efficiency, and greater resilience against multipath fading. Importantly, all variants remain backward compatible with *802.11b*, ensuring interoperability between devices of different generations [4].

Each Wi-Fi transmission is composed of frames with a well-defined structure. These frames typically include a preamble for synchronization and channel estimation, a MAC header containing addressing and control information, a payload carrying user or protocol data, and a *Frame Check Sequence (FCS)* for error detection. Of these, the preamble is especially important for radio frequency analysis because it provides a clear spectral and temporal signature that can be used to identify Wi-Fi signals in monitoring systems [4].

The prevalence of Wi-Fi makes it a major contributor to spectral occupancy in the 2.4 GHz band. It underpins everyday applications such as internet access, file transfer, video streaming, and smart home devices. In addition, several consumer drones, including the *DJI Tello*, use Wi-Fi for telemetry and control. This dual role makes Wi-Fi both relevant and challenging for RF monitoring and signal classification. On one hand, it provides an observable signature for drone detection; on the other hand, its ubiquity complicates the process of distinguishing drone transmissions from surrounding wireless activity.

2.2.2 Bluetooth (IEEE 802.15.1)

Bluetooth is a short-range wireless communication protocol that operates within the 2.4 GHz ISM band. It was originally standardized under IEEE 802.15.1 and is designed for moderate-throughput, point-to-point connections such as wireless audio, peripherals, and device interconnectivity.

Bluetooth Classic, also called Basic Rate / Enhanced Data Rate (BR/EDR), divides the 2.4 GHz band into 79 channels, each 1 MHz wide [5]. It employs frequency-hopping spread spectrum (FHSS), switching among these channels in a pseudo-random sequence at a rate of about 1,600 hops per second. This hopping mechanism helps avoid continuous interference on any single frequency and aids coexistence with other wireless technologies in the same band. Modulation schemes include Gaussian Frequency Shift Keying (GFSK) for basic rate, with $\pi/4$-DQPSK and 8-DPSK used in the higher-data rate EDR modes [5].

Bluetooth Low Energy (BLE), introduced in Bluetooth 4.0, uses a different channel scheme. BLE divides the same ISM band into 40 channels, each 2 MHz wide. Some channels are designated for advertising (used for discovery and connection setup), the rest for data exchange. The design of BLE emphasizes low power consumption and efficiency rather than continuous high throughput [5].

Because Bluetooth signals occupy much narrower bandwidth (1-2 MHz) than Wi-Fi (≈20-22 MHz), they are easier to isolate spectrally. However, the rapid hopping in Bluetooth Classic and the bursty nature of BLE transmissions make detection and classification more challenging. Signal energy is spread across many frequencies over time, which means an SDR-based system must sample widely and frequently enough to catch short transmissions on various channels.

Although Bluetooth is less dominant in terms of total spectral occupancy than Wi-Fi, its prevalence is high. Many everyday devices use Bluetooth Classic or BLE; these contribute short-duration, low-power RF activity. Thus, in any RF monitoring or signal classification system in the 2.4 GHz ISM band, Bluetooth is an essential signal class to handle and distinguish from drone or Wi-Fi traffic

2.3 Software Defined Radio (SDR)

Software Defined Radio (SDR) is a reconfigurable radio communication platform in which functions traditionally performed by hardware, such as mixing, filtering, modulation, and demodulation, are instead implemented in software. This is enabled by digitizing RF signals early in the chain with high-speed analog-to-digital converters (ADCs), allowing flexible digital signal processing [6].

An SDR typically consists of an analog front end for frequency conversion and amplification, followed by digital processing handled by a general-purpose processor (GPP), field-programmable gate array (FPGA), or digital signal processor (DSP). Protocols and waveforms are defined in software, making a single SDR capable of supporting multiple wireless standards [6].

One of the primary advantages of SDR is its flexibility and cost-effectiveness. Commercially available devices such as RTL-SDR, HackRF, and ADALM-PLUTO demonstrate how affordable SDR platforms have become, while still providing wide tuning ranges and programmable bandwidths [7]. This makes SDR practical for education, research, prototyping, and spectrum monitoring.

In the context of RF classification, SDRs are particularly powerful because they allow detailed time- and frequency-domain observation of complex environments such as the 2.4 GHz ISM band. This capability is essential for detecting and distinguishing overlapping signals like Wi-Fi, Bluetooth, and drone telemetry.

2.3.1 IQ-data Collection

IQ data refers to digitized RF signals represented by their in-phase (I) and quadrature (Q) components, which together capture both amplitude and phase information [8]. Capturing IQ data enables reconstruction and detailed analysis of the waveform in the digital domain.

In SDR systems, the RF signal is downconverted to baseband by the analog front-end. Then an analog-to-digital converter (ADC) samples the baseband signal, producing complex-valued samples where the real part is the I component and the imaginary part is the Q component. These time-domain samples preserve how the signal changes over time, before any frequency-domain transformation [8].

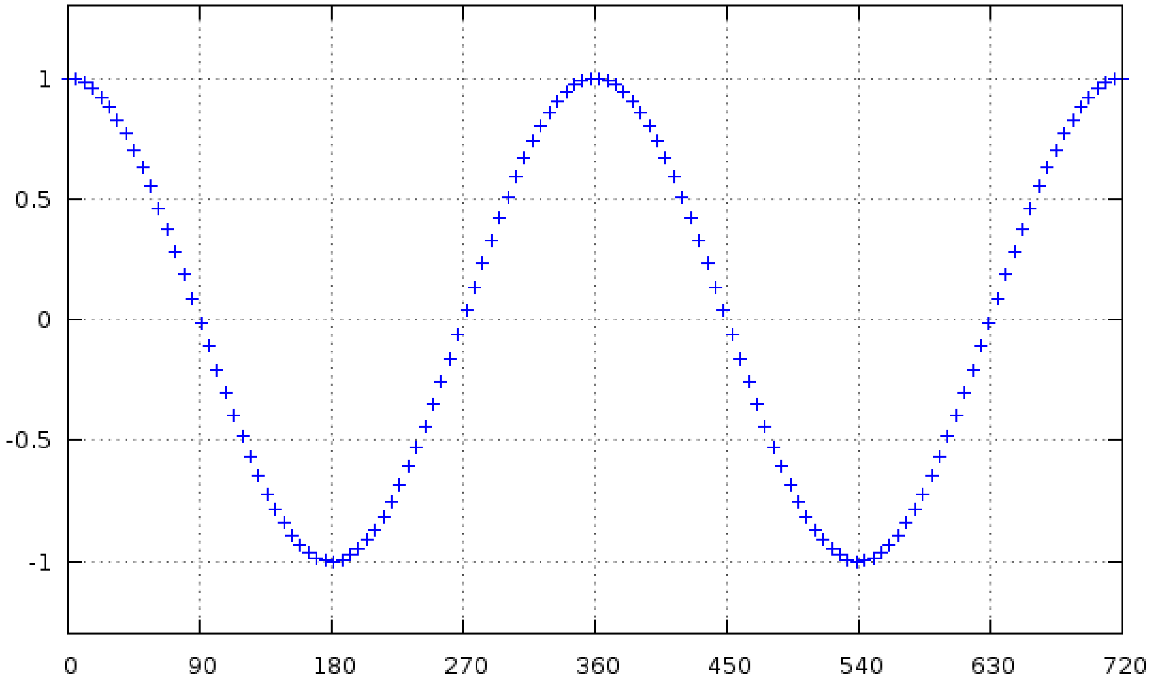

Fig. 2. Example of IQ data showing the in-phase (I) and quadrature (Q) components of a signal in the time domain [23].

Each IQ sample is a momentary snapshot encoding magnitude and phase. Collecting sequences of these samples allows computation of spectral features (for example using Fourier transforms or power spectral density) and modulation or waveform features useful for classification.

2.3.2 ADALM-PLUTO

The ADALM-PLUTO, developed by Analog Devices, is a compact and affordable SDR platform designed for educational and experimental use. It supports full-duplex transmission and reception across a tunable frequency range of 325 MHz to 3.8 GHz, enabling simultaneous transmit and receive operations. The device interfaces via USB and is powered by a Xilinx Zynq SoC, which combines an ARM processor with FPGA logic for flexible and efficient signal processing [9].

Fig. 3. The ADALM-PLUTO SDR [24]

The ADALM-PLUTO is compatible with software tools such as MATLAB, Simulink, and GNU Radio, making it accessible for a broad range of users. Its small form factor, ease of use, and real-time processing capabilities make it suitable for hands-on exploration of modern wireless systems and digital signal processing techniques [9].

TABLE I

Key Technical Specifications of the ADALM-PLUTO SDR

Parameter	Specification
RF Transceiver	ADI AD9363 (1 Tx, 1 Rx)
Frequency Range	325 MHz to 3.8 GHz
Channel Bandwidth	200 kHz to 20 MHz
Sampling Rate	65.1 kSPS to 61.44 MSPS
Data Converters	12-bit DAC/ADC
Receiver Gain Control	0 to +74.5 dB (manual, slow/fast attack)
RSSI Dynamic Range	100 dB (±2 dB accuracy)
Processing Unit	Xilinx Zynq Z-7010 SoC (ARM Cortex-A9 + FPGA)
Memory	512 MB DDR3L SDRAM, 32 MB Flash
USB Interface	USB 2.0 (up to 4 MSPS without loss)
Power Supply	USB-powered (external input optional)

Specifications adapted from [9].

The ADALM-PLUTO is based on the AD9363 transceiver but shares its architecture with the more advanced AD9364. By modifying the device via the UART interface, it can be reconfigured to match the AD9364's extended specifications, enabling a tuning range of 70 MHz to 6 GHz and a maximum channel bandwidth of 56 MHz [9][10]. This software-based modification is documented by Analog Devices and supported by the SDR community for educational and experimental use [10]. While these enhancements expand the SDR's capabilities, performance outside the default range (325 MHz to 3.8 GHz) is not guaranteed and may involve reduced sensitivity or increased noise. Nonetheless, the extended configuration provides significant value for flexible and low-cost RF experimentation.

2.3.3 MATLAB

MATLAB is a high-level programming and numeric computing environment developed by MathWorks, widely used in engineering, scientific research, and signal processing applications. It offers many built-in functions and toolboxes for tasks such as data analysis, algorithm development, and system modeling.

In the SDR context, MATLAB provides comprehensive support for signal acquisition, processing, and analysis. The Communications Toolbox and the ADALM-PLUTO support package enable interaction with SDR hardware. Users can configure SDR parameters, capture IQ data in real time, and apply algorithms for filtering, spectral analysis, and modulation classification [11].

Furthermore, MATLAB's integration with Simulink and support for live scripts enhance its usefulness for rapid prototyping and academic experimentation. These features make MATLAB an effective platform for developing, testing, and visualizing signal processing workflows in a user-friendly environment.

2.4 The frequency Spectrum

The frequency spectrum represents how a signal's power is distributed across different frequency components. In RF analysis, it is often difficult to distinguish signals in the time domain, since many appear visually similar or chaotic. Transforming signals into the frequency domain reveals distinct structures, modulation signatures, and bandwidths that are not immediately visible in the raw waveform. This transformation is commonly performed using the Discrete Fourier Transform (DFT), which provides a frequency-domain representation of a sampled signal. Such analysis is essential for identifying signal types, estimating occupied bandwidth, and detecting interference, all of which are critical tasks in RF classification systems.

2.4.1 Fast Fourier Transform (FFT)

The Discrete Fourier Transform (DFT) of a discrete signal $x[n]$ of Length N is defined as [12]:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-\frac{j2\pi kn}{N}}, \quad k = 0, 1. . ., N - 1$$

with the inverse given by:

$$x[n] = \frac{1}{N} \sum_{n=0}^{N-1} x[K] \cdot e^{-\frac{j2\pi kn}{N}}$$

Each index k corresponds to a discrete frequency bin that indicates the strength of a frequency component in the signal. While the DFT is exact, its computational complexity is $O(N^2)$, which is inefficient for large datasets. The Fast Fourier Transform (FFT) reduces this to $O(N\, log\, N)$, enabling real-time or large-scale spectrum analysis [12].

The most widely used FFT algorithm is the *Cooley–Tukey method*, which recursively decomposes a large DFT into smaller DFTs of even and odd indexed samples. This divide-and-conquer approach exploits computational symmetries and is most efficient when the sequence length N is a power of two [13].

In practical applications, the FFT is applied to a block of IQ data or time-domain samples collected from a sensor or software-defined radio (SDR). The output is a complex-valued array representing both the amplitude and phase of each frequency bin. The magnitude spectrum is typically obtained by computing the absolute value of the complex output [12][13]:

$$|X[k]| = \sqrt{(X\,[k])^2 + Im(X[k])^2}$$

For visualization and interpretation, the magnitude spectrum is often converted to a logarithmic decibel (dB) scale:

$$Magnitude(dB) = 10 \cdot log_{10}(|X[k]|^2)$$

FFT plays a fundamental role in spectrum analysis, bandwidth estimation, modulation classification, and interference detection. It allows engineers and researchers to analyze signals not just based on their time-domain appearance, but based on how energy is distributed across frequencies. This is particularly valuable in crowded spectral environments, such as the 2.4 GHz ISM band, where multiple overlapping signals may be indistinguishable in the time domain but separable in the frequency domain.

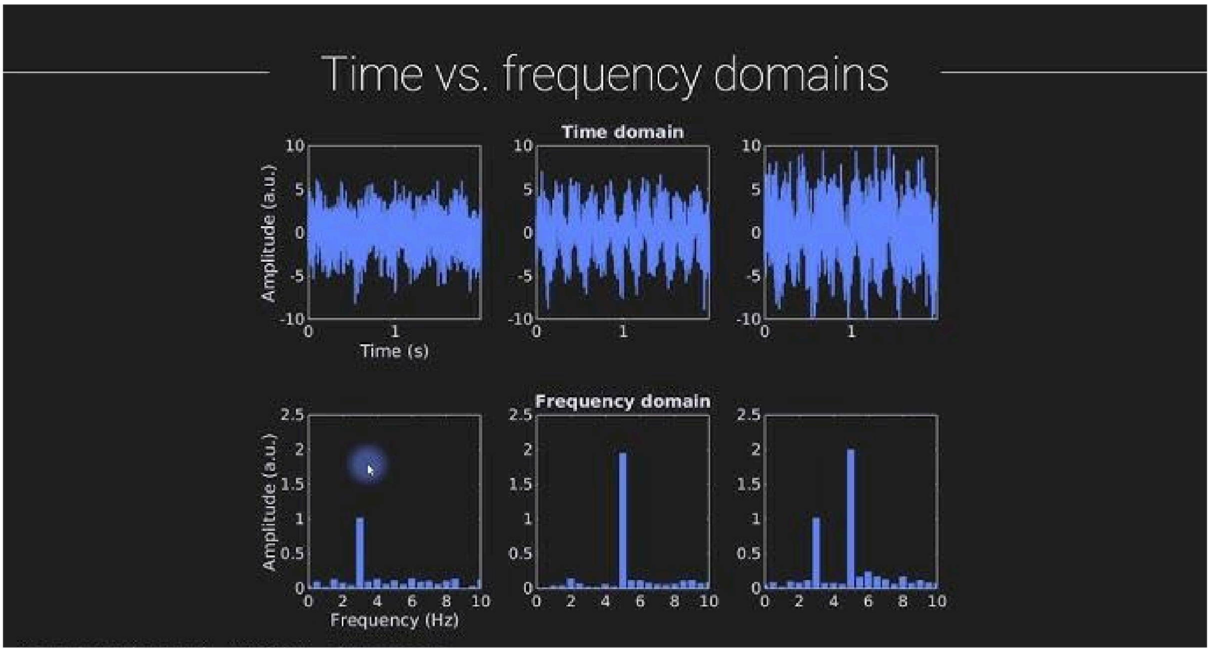

Fig. 4. Comparison of signals in the time and frequency domains, illustrating how distinct frequency components may be hidden in the time-domain waveform but revealed in the frequency domain [25]

In many cases, FFT computations are performed using software tools such as MATLAB, Python (NumPy/SciPy), or GNU Octave, which provide built-in functions to efficiently execute the algorithm on standard computing hardware.

2.4.2 Bandwidth

Bandwidth refers to the frequency range occupied by a signal; it plays a central role in signal characterization, classification, and regulatory analysis. Depending on context and application, there are several ways to define bandwidth.

Absolute bandwidth is the difference between the highest and lowest frequencies at which the signal contains energy. This measure provides a rough indication of spectral extent but can be overly affected by noise or transient components. The *3 dB bandwidth* (half-power bandwidth) instead measures the range of frequencies over which the signal power remains within half (−3 dB) of its peak. While useful in analog and filter-based systems, this definition is less reliable for burst-like or irregular signals [14].

For more robust estimation, especially in SDR and digital communication contexts, *occupied bandwidth* is commonly used. This metric defines the portion of the spectrum that contains a specified percentage of the total signal power. Two common variants are the 95 % and 99 % occupied bandwidths. The 95 % occupied bandwidth excludes a slightly larger fraction of the signal's outer energy and therefore results in a narrower measurement, while the 99 % occupied bandwidth includes nearly all of the signal power, producing a wider and more conservative value [14][15].

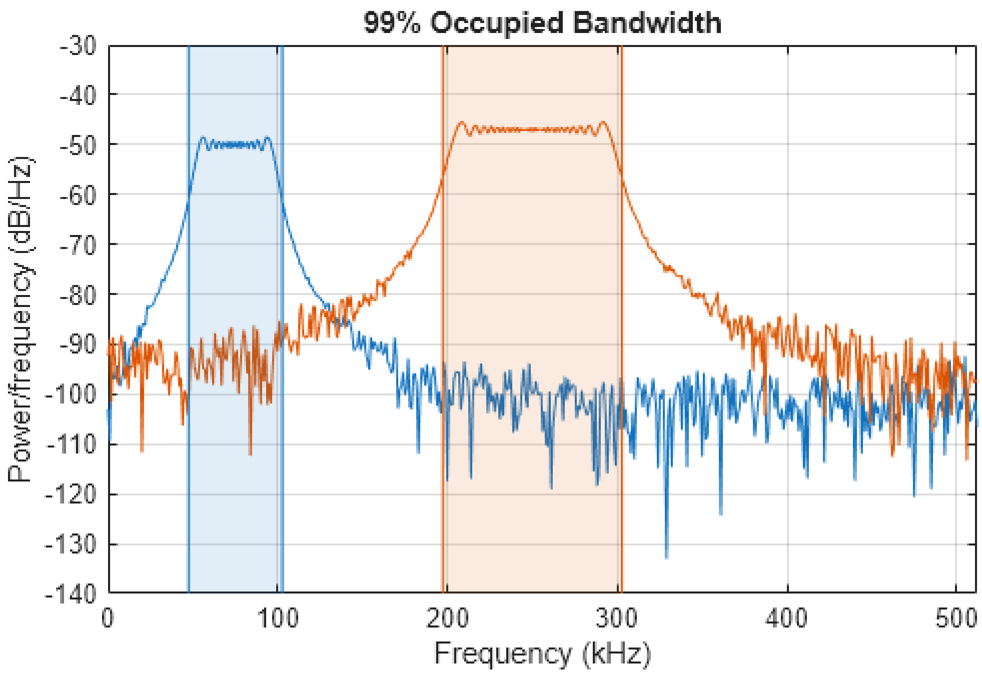

Fig. 5. Measurement examples of 99% occupied bandwidth, showing the frequency range containing 99% of the signal's total power [26].

In practice, occupied bandwidth is computed from power spectral density (PSD) estimates obtained using FFT. The total signal power is integrated across frequency bins, and the bounds are defined such that 95 % or 99 % of the power is contained within them. These definitions are widely used in both regulatory compliance and signal classification, as they provide a consistent basis for distinguishing between wideband and narrowband transmissions.

2.5 Artificial Intelligence (AI)

Artificial Intelligence (AI) refers to the design of computational systems that can perform tasks traditionally associated with human intelligence, such as reasoning, pattern recognition, and learning from data [16]. The term encompasses a wide range of approaches, from symbolic rule-based methods to modern data-driven techniques, with machine learning representing one of the most impactful developments.

This thesis focuses on those AI concepts that are directly relevant to signal classification. These include data-driven algorithms, supervised learning, neural networks, and deployment strategies. Together, they provide the theoretical basis for building systems capable of analyzing and categorizing radio signals in complex spectral environments.

2.5.1 Machine Learning (ML)

Machine learning is a branch of AI that enables systems to improve performance through experience rather than explicit programming. Models are trained by identifying patterns within data and adjusting their internal parameters to minimize prediction errors, which allows them to generalize to unseen examples [17].

Several paradigms exist, including *supervised*, *unsupervised*, and *reinforcement learning*. In supervised learning, which is most relevant to signal classification, the model is trained on labeled datasets that map inputs to known outputs. This provides a reference that allows the model to distinguish between classes and make predictions for new data.

Neural networks form one of the most widely used model classes in machine learning. They consist of interconnected layers of computational nodes, where each layer transforms its input before passing it on. Through iterative optimization, the network adjusts its weights to approximate complex non-linear functions [18].

Convolutional neural networks (CNNs) are a specialized architecture particularly effective for structured data. Convolutional filters detect local features, such as spatial patterns in images or frequency-localized structures in spectral data. This reduces the number of trainable parameters compared to fully connected networks and introduces an inductive bias aligned with how such data is structured [18]. Pooling operations are commonly used alongside convolutions to reduce dimensionality while preserving key information, improving efficiency and robustness.

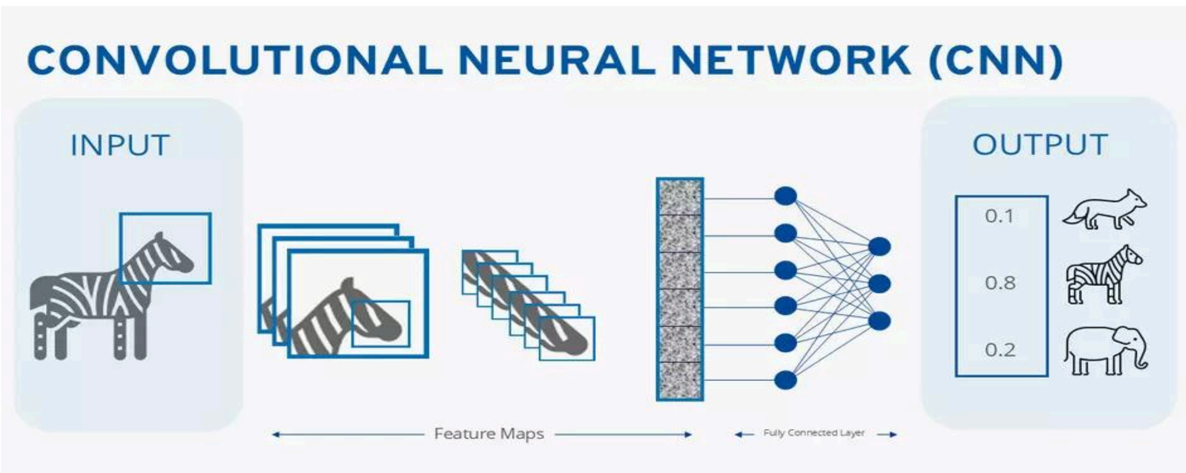

Fig. 6. Example of a convolutional neural network (CNN), where input features are extracted through convolution and pooling layers before classification in fully connected layers [27].

Training neural networks requires defining a loss function that quantifies the error between predictions and ground truth, and using optimization algorithms such as stochastic gradient descent or Adam to minimize it. Performance is then monitored using evaluation metrics such as accuracy or cross-entropy loss [18].

Machine learning has proven highly effective in domains where data is complex and analytical models are insufficient. In wireless communications, its capacity to learn non-linear patterns and classify signals in noisy environments has made it an increasingly important tool [17][18].

A more detailed workflow for the relevant machine learning process is presented in chapter "*3.3 Method II – AI-Based Signal Classification* "

2.5.2 Tools and Frameworks

The development and deployment of modern machine learning systems are supported by a wide range of software frameworks and libraries. These tools form the infrastructure for designing, training, evaluating, and operationalizing models.

One of the most widely used deep learning frameworks is *PyTorch*, which offers a dynamic computation graph, a user-friendly interface, and strong GPU acceleration. It has become the dominant research framework while also supporting large-scale production deployment [19].

Supporting libraries such as *NumPy* and *SciPy* provide efficient implementations of linear algebra, matrix operations, and numerical methods. These functions are critical for preprocessing data, performing feature extraction, and carrying out low-level computations that feed into training pipelines.

For model portability, the *Open Neural Network Exchange (ONNX)* format has emerged as a standard for exchanging models across frameworks. It enables trained models to be transferred from PyTorch to other environments, such as C++, MATLAB, or embedded devices, making it a valuable tool for deployment.

Together, these frameworks provide the computational and software backbone that allows machine learning models to move efficiently from research concepts to real-world applications.

2.5.2.1 PyTorch

PyTorch is an open-source deep learning framework developed by Meta AI. It provides the tools needed to design, train, and evaluate neural networks, making it one of the most widely used platforms for machine learning research and applications [19]. At its core, PyTorch offers a flexible programming interface where models can be defined and modified during runtime. It automatically handles gradient calculations through its autograd engine, which simplifies the implementation of backpropagation. Combined with GPU acceleration via CUDA, PyTorch enables efficient training of large-scale models.

In practice, PyTorch allows researchers and engineers to move from mathematical concepts to working implementations with relatively little overhead. It manages the computational graph, optimizes the use of hardware, and provides built-in modules for common tasks such as convolution, activation, and optimization. This makes it possible to focus on the design of learning architectures rather than low-level numerical operations.

Because of this balance between usability and performance, PyTorch has become a standard framework in both academia and industry. For tasks such as signal classification, it provides the infrastructure required to implement and train neural networks in a reliable and efficient way [19].

2.6 Ryze Tello Wi-Fi Drone

The Ryze Tello is a compact and lightweight unmanned aerial vehicle designed for educational and recreational purposes. It communicates over Wi-Fi, enabling control and telemetry through a mobile application or programming interface. The drone includes sensors that support stable flight and video capture, making it suitable for indoor operation and experimentation [20].

Fig. 7. The Ryze Tello Wi-Fi drone [28].

When powered on, the Tello creates its own Wi-Fi access point in the 2.4 GHz band using the IEEE 802.11n standard. Control commands and video streams are transmitted over this link, which allows the signal to be observable with spectrum monitoring systems. The manufacturer specifies a maximum control and video range of approximately 100 meters, although the effective range is typically shorter in real-world environments [20].

Due to its accessibility and open software development kit (SDK), the Tello is often chosen for research and prototyping in areas such as wireless communication analysis and drone detection [21].

3. Methodology

This chapter presents the implementation of two complementary methods for detecting Wi-Fi-enabled drones using software-defined radio (SDR) and signal classification techniques. It begins with the system configuration, including hardware and software components, and continues with a description of the two detection methods: a rule-based *bandwidth analysis* approach and a *machine learning*-based classifier.

3.1 System Setup

This section outlines the hardware and software required to implement and operate the detection systems. The configuration supports both bandwidth-based and machine learning-based methods, enabling signal acquisition, analysis, and classification within the 2.4 GHz ISM band.

3.1.1 Hardware Setup

Signal acquisition was performed using the ADALM-PLUTO SDR, developed by Analog Devices. The device was modified with the commonly used community “hack,” effectively extending its bandwidth to 56 MHz.

The SDR was connected to a host computer via USB and controlled using MATLAB in combination with custom Python scripts. An 18 dBi paddle antenna was employed to improve sensitivity and provide directional focus in the 2.4 GHz band. The host machine used for development and data processing was an *HP EliteBook 840 G4* running Windows 10, equipped with an *Intel Core i7 processor*. This system provided sufficient computational resources for real-time spectrum analysis, data logging, and neural network training. While this specific laptop was used during testing, any modern system with similar specifications and USB 2.0 or higher connectivity is suitable.

The target signal source was the *Ryze Tello* drone, which transmits over Wi-Fi using the 802.11n standard. Once powered on, the drone creates a local Wi-Fi access point and begins streaming video when active. This results in a consistent and easily detectable RF footprint within the 2.4 GHz band. Notably, the drone autonomously selects an available Wi-Fi channel at startup based on the prevailing interference environment and initiates signal transmission without requiring the drone to be airborne.

Fig. 8. Hardware setup including Windows laptop, ADALM-PLUTO SDR with 18 dBi paddle antenna, and Ryze Tello drone.

3.1.2 Software Environment

MATLAB was used to interface with the SDR, control channel sweeping, perform real-time FFT analysis, and visualize signal power distributions. Custom scripts automated the scanning process and applied bandwidth-based detection thresholds. The PlutoSDR MATLAB support package provided control of sampling rate, frequency tuning, and gain.

Python and PyTorch were used for the machine learning pipeline. NumPy and SciPy supported data manipulation, while PyTorch handled model definition, training, and evaluation. Custom scripts managed PSD sample labeling, normalization, and ONNX export. Development was carried out in *Visual Studio Code*.

Several MATLAB toolboxes were required:

- *Communications Toolbox* — for modulation, spectral estimation, and feature extraction (e.g., *pwelch*).
- *Communications Toolbox Support Package for ADALM-PLUTO Radio* — for SDR control and configuration.
- *Signal Processing Toolbox* — for filtering, windowing, and FFT operations.
- *MATLAB Coder* (optional) — for exporting ONNX models to external platforms.
- *Deep Learning Toolbox* (optional) — for compatibility testing with MATLAB-based neural networks.

Together, these tools provided a modular environment that supported both traditional signal processing and AI-based classification workflows.

3.2 Method I – Bandwidth-Based Detection

The first detection method distinguishes between different signal types in the 2.4 GHz ISM band by measuring their occupied bandwidth. Wireless devices exhibit characteristic bandwidths: Wi-Fi transmissions typically occupy 20–22 MHz, while Bluetooth signals are much narrower at around 1–2 MHz. By comparing the observed bandwidth against these known values, transmissions can be classified according to the decision process illustrated in Figure 9. Signals that do not fit either profile are flagged as potential drone activity.

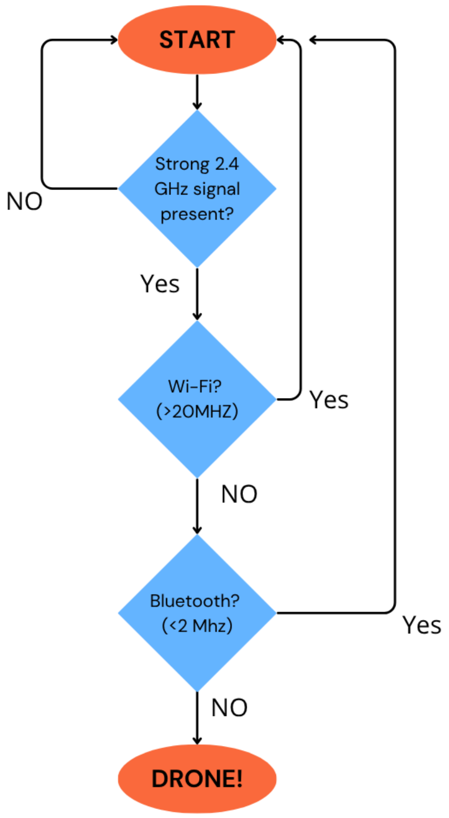

Fig. 9. Flowchart of the bandwidth-based signal classification method for distinguishing drone transmissions from conventional Wi-Fi and Bluetooth.

To establish a reference for the Ryze Tello drone, a spectral fingerprint was collected. Using MATLAB together with the ADALM-PLUTO support package, IQ data was captured while the drone transmitted video in a stationary mode. Built-in FFT functions were applied to transform the raw samples into the frequency domain, and the 99 % occupied bandwidth (OBW) was calculated. In repeated tests, the Tello signal consistently measured around 16.50 MHz, with minor fluctuations of approximately ±0.5 MHz depending on distance, but always remaining within the *16–17 MHz range*.

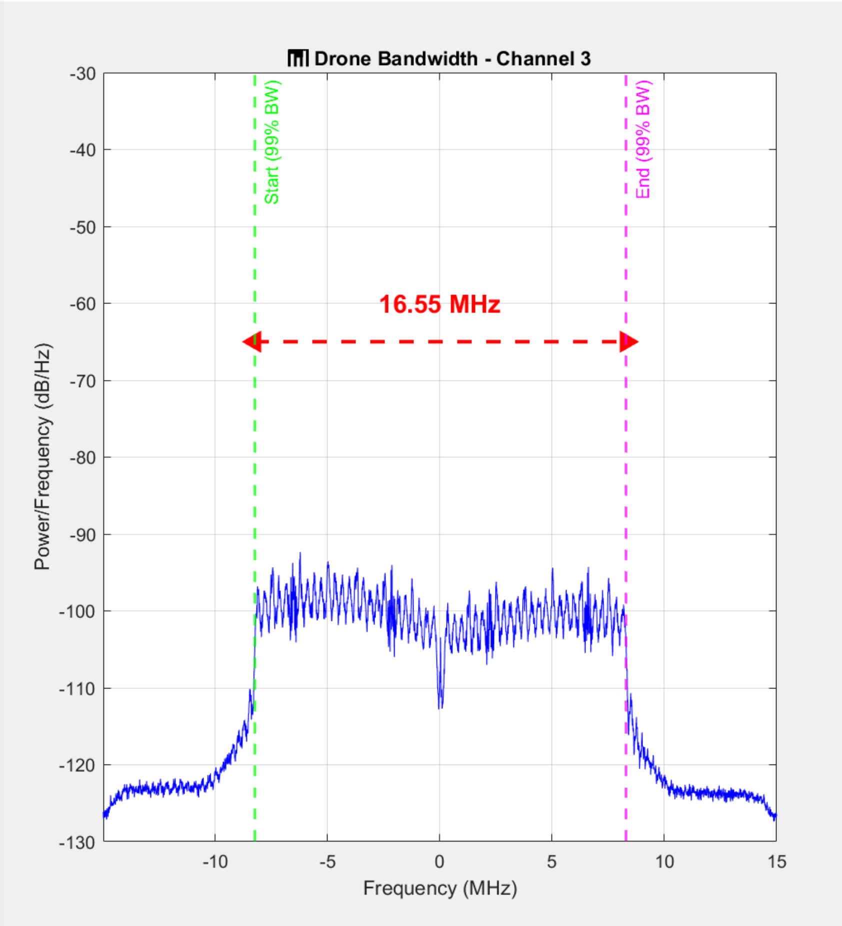

Fig. 10. Measured bandwidth of the Tello drone signal on channel 3, showing a 99% occupied bandwidth of approximately 16.5 MHz.

Based on this fingerprint, a rule-based classification procedure was defined (similar to Figure 9). The system sweeps the 2.4 GHz band by stepping through the Wi-Fi channels (1–11). At each center frequency, a short segment of IQ data is captured and transformed into the frequency domain. A manual threshold is applied to distinguish signals from background noise. If a valid signal is present, its bandwidth is estimated. Signals wider than 20 MHz are classified as Wi-Fi, those narrower than 2 MHz as Bluetooth, and signals within the 16–17 MHz range are flagged as drone transmission, a criterion derived specifically from measurements of the DJI Tello.

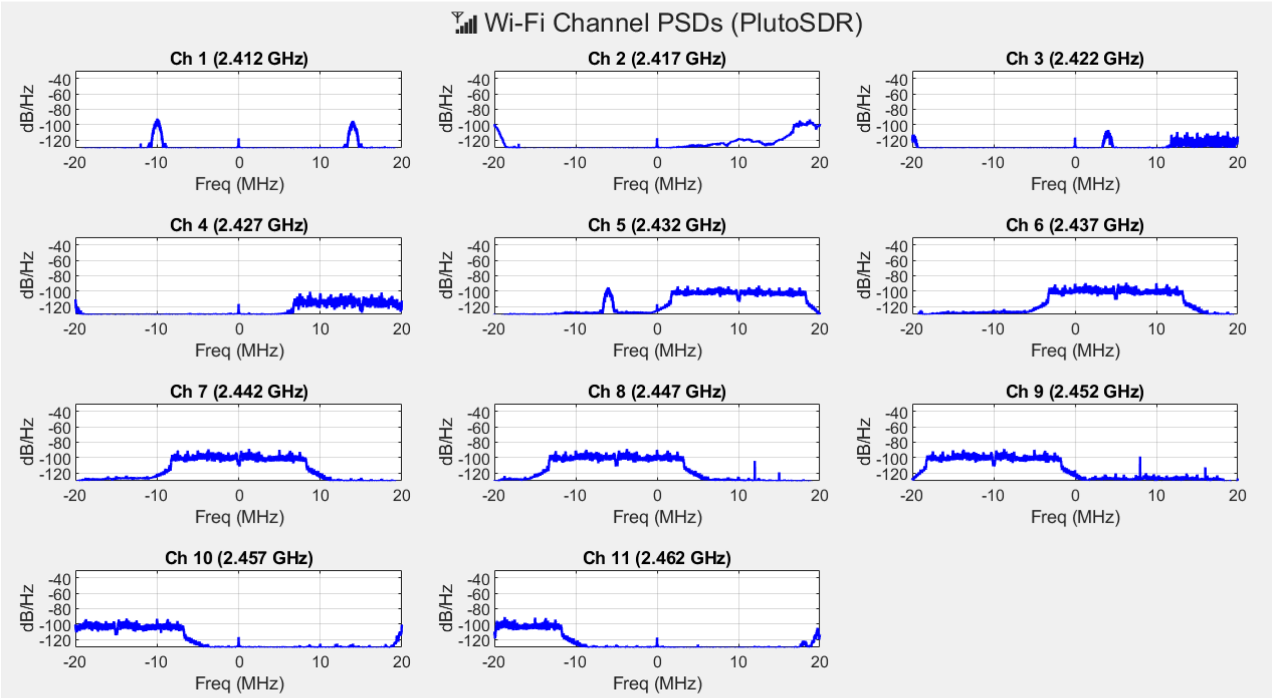

Fig. 11. Power spectral densities (PSDs) of Wi-Fi channels 1 to 11 measured with the ADALM-PLUTO SDR during a channel sweep.

Figure 11 illustrates the power spectral densities (PSDs) of Wi-Fi channels 1 to 11, obtained using the PlutoSDR during a channel sweep. The detected signal occupies a bandwidth of approximately 16.5 MHz, which is consistent with drone activity. Because this bandwidth is wider than the 5 MHz spacing between adjacent IEEE 802.11 channels, the transmission overlaps significantly with neighboring channels - a well-documented property of Wi-Fi signals in the 2.4 GHz band [4]. This explains why the same square-shaped signal appears across multiple channels.

This overlap can be utilized in order to determine the specific channel used for transmission by applying an algorithm that identifies the channel whose power spectral density is most symmetrically distributed around its center frequency. In the example shown in Figure 11, channel 7 demonstrates the most symmetric PSD shape, suggesting that 2.442 GHz (channel 7) is the center frequency used by the transmitting device. This method allows for the identification of the primary transmission channel, even when overlap is present across multiple adjacent channels.

3.3 Method II – AI-Based Signal Classification

The same measurement setup as before was used to capture signal samples, with the SDR and MATLAB FFT providing power spectral density (PSD) estimates. Three signal types were recorded for comparison:

- **Conventional Wi-Fi** from a household Wi-Fi-router.
- **DJI Tello drone** while transmitting live video to mobile phone
- **Bluetooth** while transmitting stored files between a laptop and a mobile phone.

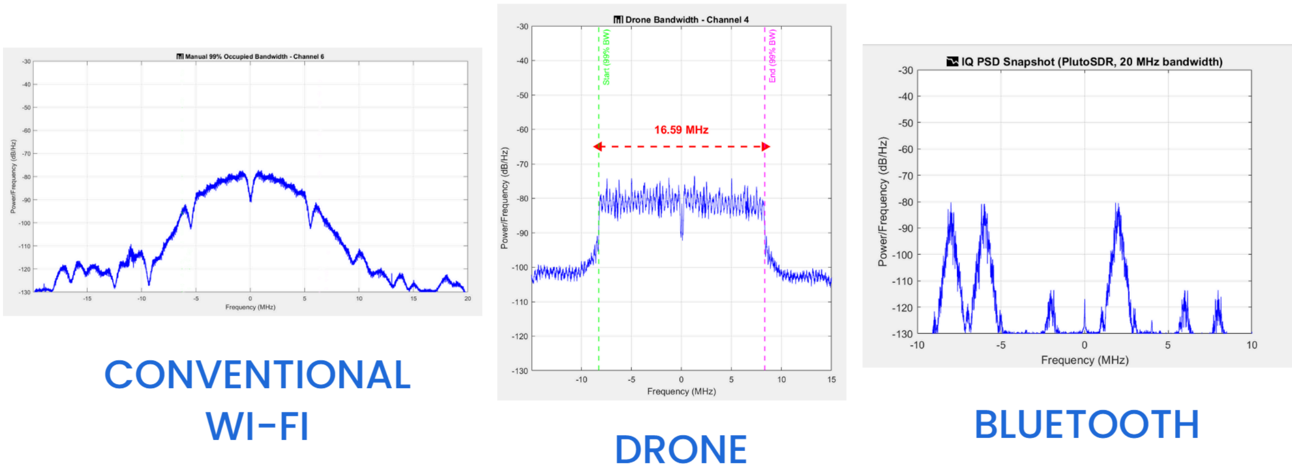

Fig. 12. Comparison of spectral shapes for conventional Wi-Fi, drone, and Bluetooth signals.

When placed side by side in the frequency spectrum, these signals show not only different bandwidths but also different *shapes* (see figure 12). Wi-Fi transmissions have a rounded spectral form, the drone produces a flatter square-like profile, and Bluetooth appears as short, bursty spikes.

This method aims to distinguish drone signal emissions from other signal types in the 2.4 GHz band by analyzing the spectral waveform and classifying it even when mixed with background noise or when the signal is weak in amplitude and would normally be regarded as noise.

3.3.1 Dataset Generation

The machine learning model used in this project is *supervised*, which means it requires a large labeled dataset of examples before it can make reliable classifications on unseen signals. To create this dataset, a custom MATLAB program was developed to record and save frequency sweeps using the SDR.

Each sweep covered Wi-Fi channels 1 to 11. For every channel, a power spectral density (PSD) estimate was calculated using FFT-based methods, resulting in a one-dimensional vector of 100 frequency bins. The full sweep was then assembled into a two-dimensional sample of size [11 × 100], where each row represented one channel.

To enable supervised learning, each sample was labeled at the time of recording:

1 for drone emissions (active or idle) and

0 for all other cases, including Wi-Fi and Bluetooth transmissions, or background noise

The dataset was collected under varied conditions to capture environmental and noise diversity. For the *drone class*, a total of *300 samples* were recorded:

- 100 indoor long-range (> 15 m)
- 100 indoor short-range (≤ 15 m)
- 100 outdoor at approximately 20 m

In addition, *200 non-drone samples* were collected to represent background activity and interference:

- 100 indoor (Wi-Fi, Bluetooth, idle/no signal, mixed)
- 100 outdoor (same variations)

In total, the dataset consisted o*f 500 labeled samples*, covering both drone and non-drone signals under varied real-world conditions. These were saved in a structured format (.npy), making them directly compatible with PyTorch’s dataset pipeline and sufficient to train a small but reliable model.

3.3.2 Model Training

The training process followed a standard supervised learning approach. The labeled samples and their classes were loaded into PyTorch, where they were normalized and grouped into small batches for processing. Much of this setup, such as batching and normalization, was handled automatically by PyTorch's built-in functions.

A lightweight convolutional neural network (CNN) was used as the model. It consisted of a few convolutional layers to detect patterns in the spectral data, followed by fully connected layers that produced a probability score for whether a signal was from a drone or not.

The model was trained over 20 rounds, gradually adjusting its internal weights to reduce errors. During training, 80% of the dataset was used for learning and 20% was kept aside for validation. This allowed performance to be monitored on unseen data, ensuring that the model was learning general patterns rather than memorizing the training examples.

Finally, the trained model was exported to the *ONNX format*, an open standard that makes it possible to run the classifier (trained model) outside Python in environments such as MATLAB or embedded systems. This allowed the model to be reused directly in the real-time detection workflow without needing to be rewritten.

3.3.3 Application and Live Inference

With the trained model exported to ONNX format, it was integrated into a MATLAB-based signal scanning system for real-time use. The SDR continuously swept across Wi-Fi channels 1 through 11, capturing fresh IQ samples at each step. For every channel, a power spectral density (PSD) estimate was computed and stored in the same format as the training data: a two-dimensional array with 11 rows and 100 frequency bins per channel.

These PSD arrays were grouped to form a complete sample and normalized using the same parameters applied during training. Each sample was then passed to the ONNX model, which returned a probability score between 0 and 1 representing the likelihood of drone activity.

A confidence threshold of 0.95 (95%) was implemented to ensure reliable detections. When the model's predicted probability exceeded this threshold, a detection event was triggered. The system subsequently recorded the occurrence and reported the model's confidence level in the classification.

4. Results

4.1 Bandwidth analysis

While the system was running, the drone was turned on, connected to a mobile phone and lifted off the ground (optional). The SDR performed continuous sweeps across the 2.4 GHz band, with each sweep taking approximately two seconds. A detection was typically achieved within one to three sweeps, corresponding to a delay of about 2–6 seconds after the drone was successfully connected to the mobile device. In outdoor tests, the maximum reliable range detection was approximately *10 meters*, occasionally more in favorable conditions.

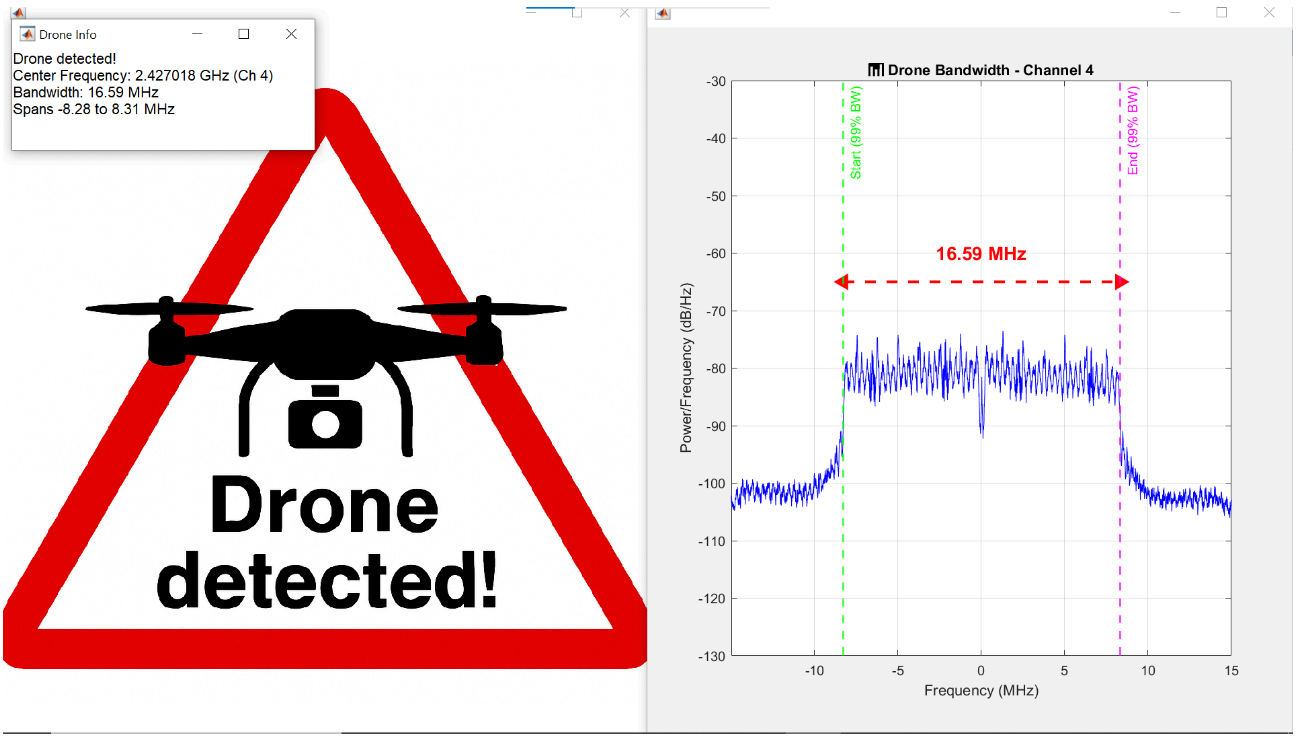

Fig. 13. Displayed output after detection of drone

Once a signal is flagged, an alert window appears as shown in Figure 13. The alert displays a live snapshot of the detected signal on the transmitted channel. On the graph, the measured bandwidth is marked, and a separate pop-up box shows the exact bandwidth, span, and center frequency, corresponding to the Wi-Fi channel used by the drone.

4.2 AI Classification

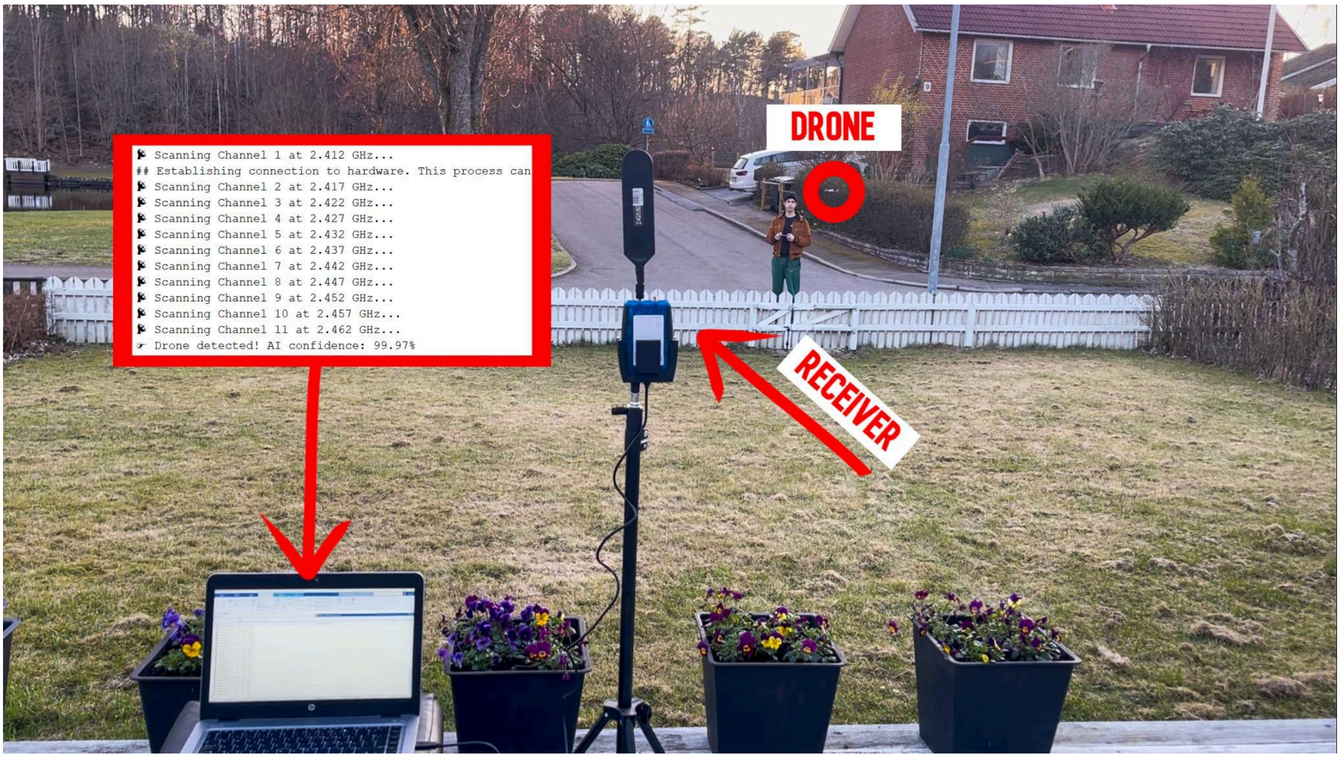

Fig. 14. Experimental setup for AI-based drone detection, showing the SDR receiver, drone under test, and connected laptop running the classifier

While the system is running, the SDR continuously sweeps across all 11 Wi-Fi channels and the captured data was evaluated by the AI classifier. When the drone was connected to the a mobile device within range, the system returns a confidence value close to 100%

```
📡 Scanning Channel 1 at 2.412 GHz...
## Establishing connection to hardware. This process can take several seconds.
📡 Scanning Channel 2 at 2.417 GHz...
📡 Scanning Channel 3 at 2.422 GHz...
📡 Scanning Channel 4 at 2.427 GHz...
📡 Scanning Channel 5 at 2.432 GHz...
📡 Scanning Channel 6 at 2.437 GHz...
📡 Scanning Channel 7 at 2.442 GHz...
📡 Scanning Channel 8 at 2.447 GHz...
📡 Scanning Channel 9 at 2.452 GHz...
📡 Scanning Channel 10 at 2.457 GHz...
📡 Scanning Channel 11 at 2.462 GHz...
👉 Drone detected! AI confidence: 99.97%
```

Fig. 15. Close-up of the AI detection output, indicating successful drone detection with a confidence score above 99%.

As shown in Figure 14, once the confidence level passes the 95% threshold, a message appears stating *“Drone detected!”* together with the exact confidence score. At this point, the sweeping process stops, confirming the detection.

Detection was usually done within one or two sweeps, making the effective detection time between **2–4 seconds**, or slightly longer if the drone was further away. In outdoor tests, the maximum reliable range was approximately **20 meters**.

5. Discussion

This chapter examines the performance of the implemented detection methods, outlines their limitations, and explains the motivation for introducing a complementary approach. In addition, directions for future work are discussed

5.1 Motivation for a Second Method

The first detection method was based on a fixed power threshold. In practice, this required manually setting a cutoff level in dB to separate background noise from valid signals. While effective at short range, the method struggled at longer distances, where parts of the drone signal could fall below the threshold and be missed. Conversely, if the threshold was set too low, random noise fluctuations could be misinterpreted as valid signals.

In some cases, wide noise bursts or overlapping interference could appear to have a bandwidth of 16–17 MHz, causing the system to incorrectly flag them as drone activity. This problem was mostly mitigated by additional MATLAB filtering, but it could still occur occasionally under unfavorable conditions.

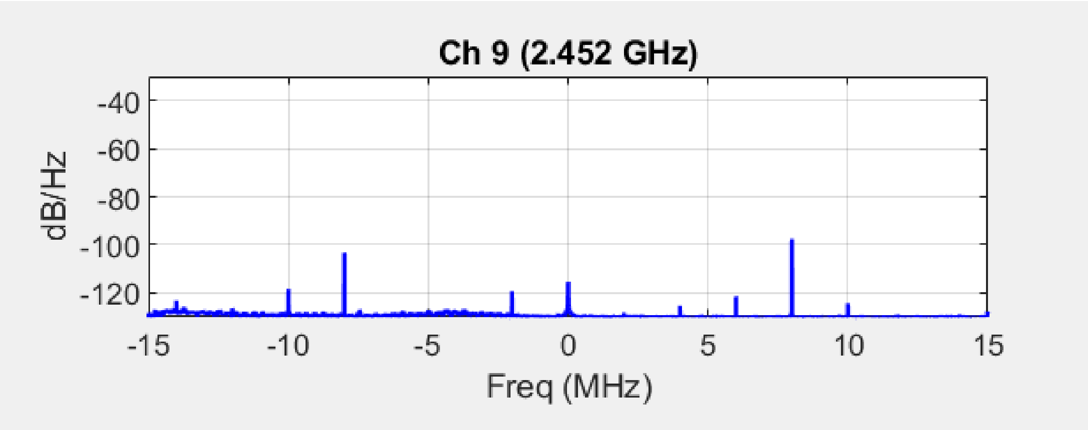

Fig. 16. On channel 9, two smaller noise spikes at –8 MHz and +8 MHz were interpreted as having a combined bandwidth of approximately 16 MHz when the detection threshold was set too low, resulting in a false trigger. In reality, the drone was transmitting on channel 4 (see Fig. 17).

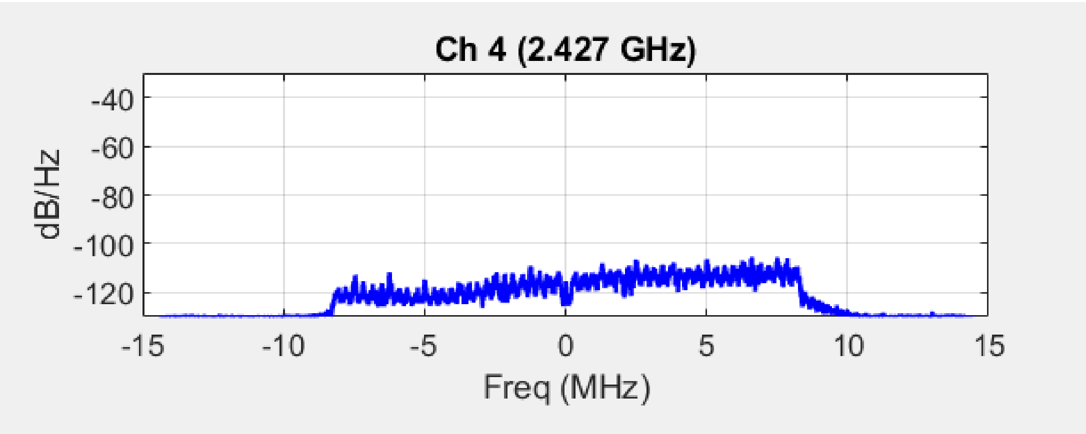

Fig. 17. Actual drone signal measured on channel 4. Because the signal level was lower than the noise spikes on channel 9, it was overlooked by the threshold-based system and therefore missed.

Another limitation appeared when signals overlapped during sweeps. For example, a Bluetooth burst near the drone's transmission could distort the measured spectrum, creating extended "tails" that made the bandwidth appear much wider. In such cases, the system often classified the signal as conventional Wi-Fi instead of a drone.

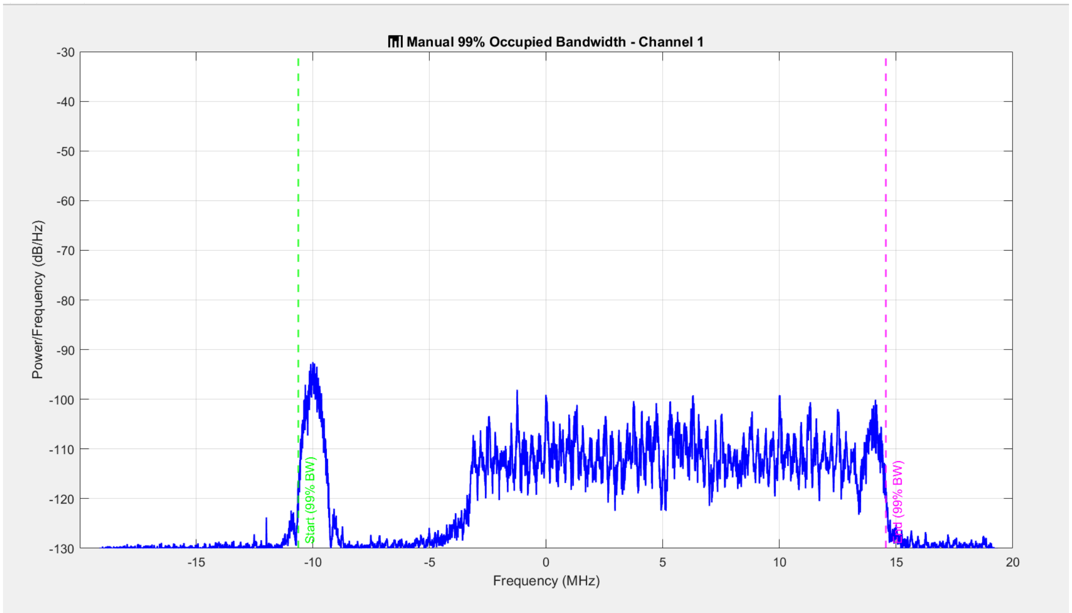

Fig 18. Drone signal mistakenly measured together with a nearby Bluetooth burst, resulting in an incorrect measured bandwidth of about 25 MHz.

While both of these issues were generally easy for a human observer to recognize by looking at the spectral shape, the threshold-based method lacked the flexibility to make such distinctions. This motivated the development of a second method, based on artificial intelligence, which could detect drone emissions by learning to recognize their square spectral shape rather than relying on absolute thresholds. By doing so, the system became more robust against noise, interference, and varying signal amplitudes, much like how a human would visually identify a drone signal.

Both detection methods performed very accurately and quickly during indoor tests. This was likely due to signal reflections from walls, which extended the effective range, and the lower level of interference compared to outdoor environments. The AI-based method was consistently faster than the bandwidth method, particularly when additional noise sources were introduced near the receiver. Neither system produced false triggers very often, which indicates that both approaches were stable under controlled conditions.

The biggest difference between the two methods appeared in outdoor tests. The bandwidth-based method began to struggle after 10 meters, as the signal power dissipated more quickly without walls to reflect the signal, and possible interference made detection less reliable. By contrast, the AI-based method maintained reliable detection outdoors and was able to classify drone signals more quickly at ranges up to about 20 meters.

Although this outdoor range may appear low, it is not a limitation of the detection system itself but rather of the drone hardware. While DJI specifies that the Tello drone should operate at distances up to 100 meters, practical testing revealed otherwise. Beyond 15 meters, the drone already displayed weak signal warnings, and at 20–25 meters it consistently lost connection. This observation is consistent with independent reviews of the drone online and reflects the fact that the DJI Tello is an entry-level, short-range drone.

5.2 Comparison Between Bandwidth and AI-Based Detection

Both detection methods performed very accurately and quickly during indoor tests. This was likely due to signal reflections from walls, which extended the effective range, and the lower level of interference compared to outdoor environments. The AI-based method was consistently faster than the bandwidth method, particularly when additional noise sources were introduced near the receiver. Neither system produced false triggers very often, which indicates that both approaches were stable under controlled conditions.

The biggest difference between the two methods appeared in outdoor tests. The bandwidth-based method began to struggle after 10 meters, as the signal power dissipated more quickly without walls to reflect the signal, and possible interference made detection less reliable. By contrast, the AI-based method maintained reliable detection outdoors and was able to classify drone signals more quickly at ranges up to about 20 meters.

Although this outdoor range may appear low, it is not a limitation of the detection system itself but rather of the drone hardware. While DJI specifies that the Tello drone should operate at distances up to 100 meters, practical testing revealed otherwise. Beyond 15 meters, the drone already displayed weak signal warnings, and at 20–25 meters it consistently lost connection. This observation is consistent with independent reviews of the drone online and reflects the fact that the DJI Tello is an entry-level, short-range drone.

Although the AI model was ultimately faster and more accurate, it required significant effort to reach that level of reliability. A large number of samples had to be collected, and the model needed careful fine-tuning before it could outperform the bandwidth-based method. In the early stages, when only a small dataset was available, the AI method was highly unreliable, especially when samples collected in one location were tested in a different environment.

By contrast, the bandwidth-based method is much easier to build and deploy. It does not require any training data and can therefore be applied directly in new or “untrained” environments. While less robust overall, its simplicity makes it useful in situations where data collection and model training are not feasible.

5.3 Confidence Threshold in AI Classification

An important factor in the AI-based method is the confidence threshold, which can be manually adjusted to balance speed and reliability. A lower threshold, for example *70%,* enables faster detections since the system triggers earlier, but it also increases the risk of false positives if interfering signals are misclassified as drones. A higher threshold, such as *95–99%*, provides stronger certainty but requires more sweeps before triggering, which slows down the response.

The choice of threshold depends on the application. In critical environments, such as military or high-security areas, it may be preferable to accept more false alarms rather than risk missing a true detection. In less sensitive environments, however, a higher threshold may be more appropriate to minimize false triggers.

5.4 Conclusion

Although the drone used in this study was limited to a practical control range of about 20 meters, two detection systems were successfully implemented that could reliably identify and classify its signals within this range. Both systems operated in real time and were built using a relatively inexpensive and accessible software-defined radio.

5.5 Future Work

Future studies should include a wider range of drone models to determine whether the observed bandwidth of 16–17 MHz and the square-shaped spectral profile are unique to the DJI Tello or representative of a broader pattern among Wi-Fi–based drones. Expanding the dataset with different platforms would strengthen the generalizability of the AI model and reduce the risk of overfitting to a single drone type.

It would also be important to test the detection system over longer distances. Both SDR hardware and many drone platforms are capable of operating across several kilometers, far beyond the 20-meter range of the DJI Tello. Evaluating system performance under such conditions would provide valuable insight into scalability and the feasibility of deploying low-cost SDR-based detection systems in real-world scenarios.

Another relevant direction for future research is to assess the system's performance on Wi-Fi–enabled drones that do not transmit live video but rely solely on control communication. These platforms may exhibit narrower bandwidth usage and less distinctive spectral features compared to video-streaming drones like the DJI Tello. Testing against this category of drones would help clarify whether the detection methods remain effective under reduced traffic conditions, or if additional feature engineering and model adaptation are necessary.

Lastly, it would be worthwhile to investigate scenarios where the spectral characteristics of drone communication, such as bandwidth and spectral shape, more closely resemble conventional Wi-Fi traffic or vary substantially across drone models. In such cases, the current methods may face challenges in generalization, underscoring the need for more robust and adaptable detection strategies. Future research could therefore focus on developing techniques that balance broad applicability across diverse drone types with sufficient sensitivity to capture unique communication signatures, thereby avoiding overly specialized solutions tailored to a single platform.

6. Reference list

[1] Federal Communications Commission, *Part 15 – Radio Frequency Devices*, Title 47 of the Code of Federal Regulations (CFR), Washington, DC, USA.

[2] International Telecommunication Union, *Radio Regulations – Article 5: Frequency Allocations*, Geneva, Switzerland, 2020.

[3] A. Kamerman and L. Monteban, "WaveLAN-II: A High-Performance Wireless LAN for the Unlicensed Band," *Bell Labs Technical Journal*, vol. 2, no. 3, pp. 118–133, 1997.

[4] IEEE, *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*, IEEE Std 802.11-2020, Feb. 2021.

[5] "Comparison of Bluetooth BR/EDR and Bluetooth LE Specifications," *MathWorks Help*, accessed [Accessed: Sep. 21, 2025], available: https://www.mathworks.com/help/bluetooth/gs/comparison-of-bluetooth-bredr-and-bluetooth-le.html

[6] R. Akeela and B. Dezfouli, "Software-defined Radios: Architecture, State-of-the-art, and Challenges," *arXiv preprint*, Apr. 18, 2018.

[7] A. Tato, "Software Defined Radio: A Brief Introduction," *Proceedings*, MDPI, 2018.

[8] T. Zieliński, "Introduction to SDR: IQ Signals and Frequency Up-Down Conversion," in *Starting Digital Signal Processing in Telecommunication Engineering*, AGH University of Krakow, 2021, pp. 483-515.

[9] Analog Devices, "ADALM-PLUTO SDR Technical Specifications," [Online]. Available: https://wiki.analog.com/university/tools/pluto/devs/specs

[10] Analog Devices, "ADALM-PLUTO SDR Hackers Guide," [Online]. Available: https://wiki.analog.com/university/tools/pluto/hackers; see also "ADALM-PLUTO SDR Hack: Tune 70 MHz to 6 GHz," *RTL-SDR Blog*, [Online]. Available: https://www.rtl-sdr.com/adalm-pluto-sdr-hack-tune-70-mhz-to-6-ghz-and-gqrx-install/

[11] "ADALM-PLUTO Radio Support from Communications Toolbox," MathWorks, accessed Sept. 20, 2025. Available: https://www.mathworks.com/hardware-support/adalm-pluto-radio.html

[12] K. Phani Srinivas and P. S. Aithal, "Practical Oriented Analysis on Signal Processing Using FFT Algorithm," *International Journal of Electronics Engineering and Applications*, vol. 8, no. II, pp. 1-10, 2020.

[13] "Mixed-Signal and DSP Design Techniques, Fast Fourier Transforms," Analog Devices, technical handbook.

[14] ECC, *ECC Recommendation (06)01 – Bandwidth, Range and Occupied Bandwidth Measurement Methods*, Electronic Communications Committee (within CEPT), May 2006.

[15] Y. S. Kim, “A Practical Measurement Method of the Occupied Bandwidth at Monitoring Stations,” *Journal of Radio Engineering*, vol. 28, no. 3, pp. 329-341, 2019.

[16] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, 4th ed., Pearson, 2021.

[17] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer, 2006.

[18] Y. LeCun, Y. Bengio, and G. Hinton, “Deep Learning,” *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[19] Paszke *et al.* — “PyTorch: An Imperative Style, High-Performance Deep Learning Library”

[20] Ryze Tech, “Tello Specifications,” Ryze Robotics, 2025. [Online]. Available: https://www.ryzerobotics.com/tello/specs

[21] A. Dalmau Oliver, “Drone communication analysis using SDR and GNU Radio: DJI Tello,” Bachelor Thesis, Universitat Politècnica de Catalunya, 2020. [Online]. Available: https://upcommons.upc.edu/handle/2117/333032

Figure Sources

[22] Nordic Semiconductor, “What is Wi-Fi?,” *Wi-Fi Fundamentals – Lesson 1*, Academy, [Online]. Available: https://academy.nordicsemi.com/courses/wi-fi-fundamentals/lessons/lesson-1-wifi-fundamentals/topic/what-is-wifi/. [Accessed: Sep. 20, 2025].

[23] PE0SAT, “IQ Data explained,” *PE0SAT Amateur Radio and Satellite Communication*, [Online]. Available: https://www.pe0sat.vgnet.nl/sdr/iq-data-explained/. [Accessed: Sep. 20, 2025].

[24] Analog Devices, “ADALM-PLUTO: Active Learning Module for SDR,” *Analog Devices*, [Online]. Available: https://www.analog.com/en/resources/evaluation-hardware-and-software/evaluation-boards-kits/adalm-pluto.html#eb-overview. [Accessed: Sep. 20, 2025].

[25] M. X. Cohen, “Time and frequency domains,” *YouTube*, Jun. 17, 2019. [Online]. Available: https://www.youtube.com/watch?v=fYtVHhk3xJ0. [Accessed: Sep. 20, 2025]

[26] MathWorks, “obw,” *Signal Processing Toolbox Documentation*, [Online]. Available: https://www.mathworks.com/help/signal/ref/obw.html. [Accessed: Sep. 21, 2025].

[27] IONOS, “Convolutional Neural Networks (CNN): definition and functioning,” *IONOS Digital Guide*, [Online]. Available: https://www.ionos.fr/digitalguide/sites-internet/developpement-web/convolutional-neural-networks/. [Accessed: Sep. 21, 2025].

[28] DJI Stockholm, “DJI Ryze Tello,” *DJI Stockholm*, [Online]. Available: https://djistockholm.se/product/tello/dji-ryze-tello/#lg=1&slide=5. [Accessed: Sep. 21, 2025].