RIDS: Radio intrusion detection and diagnosis system for wireless communications in smart environment

Pierre-François Gimenez, Jonathan Roux, Eric Alata, Guillaume Auriol, Mohamed Kaâniche, Vincent Nicomette

RIDS: Radio intrusion detection and diagnosis system for wireless communications in smart environment

PIERRE-FRANCOIS GIMENEZ, JONATHAN ROUX, ERIC ALATA, GUILLAUME AURIOL, MOHAMED KAANICHE, and VINCENT NICOMETTE, LAAS–CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France

The expansion of the Internet-of-Things (IoT) market is visible in homes, factories, public places, and smart cities. While the massive deployment of connected devices offers opportunities to improve quality of life and to develop new services, the impact of such devices on the security of the users in a context where the level of malicious threat continues to increase is a major concern. One of the challenges is the heterogeneity and constant evolution of wireless technologies and protocols used. To overcome this problem, we propose RIDS, a *Radio Intrusion Detection System* that is based on the monitoring and profiling of radio communications at the physical layer level using autoencoder neural networks. RIDS is independent of the wireless protocols and modulation technologies used. Besides, it is designed to provide a threefold diagnosis of the detected anomalies: temporal (start and end date of the detected anomaly), frequential (main frequency of the anomaly), and spatial (location of the origin of the anomaly). To demonstrate the relevance and the efficiency of our approach, we collected a large dataset of radio-communications recorded with three different probes deployed in an experimental room. Multiple real-world attacks involving a wide variety of communication technologies are also injected to assess the detection and diagnosis efficiency. The results demonstrate the efficiency of RIDS in detecting and diagnosing anomalies that occurred in the 400–500 Mhz and 800–900 Mhz frequency bands. It is noteworthy that compromised devices and attacks using these communication bands are generally not easily covered by traditional solutions.

Additional Key Words and Phrases: IoT, IDS, software-defined radio, deep learning, wireless security

ACM Reference Format:
Pierre-Francois Gimenez, Jonathan Roux, Eric Alata, Guillaume Auriol, Mohamed Kaaniche, and Vincent Nicomette. 2019. RIDS: Radio intrusion detection and diagnosis system for wireless communications in smart environment. *ACM Transactions on Cyber-Physical Systems* 1, 1 (January 2019), 27 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Wireless devices such as speakers, TVs, cameras, doors, shutters, light bulbs, and sensors are now widely used in our daily lives. These connected devices have specific characteristics that distinguish them from traditional devices. For example, 1) their communications are usually distributed rather than centralized through an Internet router, 2) they have limited resources and are nomadic, making it difficult to contain the effects of potentially malicious behavior if they are compromised, 3) they use various and often proprietary protocols and technologies, and are subject to frequent changes in communication technologies and the environment, 4) the users may not be aware that some devices

Authors' address: Pierre-Francois Gimenez, pierre-francois.gimenez@laas.fr; Jonathan Roux, jonathan.roux@laas.fr; Eric Alata, eric.alata@laas.fr; Guillaume Auriol, guillaume.auriol@laas.fr; Mohamed Kaaniche, mohamed.kaaniche@laas.fr; Vincent Nicomette, vincent.nicomette@laas.frLAAS–CNRS, Université de Toulouse, CNRS, INSA, Toulouse, France.

XXXX-XXXX/2019/1-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

have wireless communication capabilities (e.g., a wireless water meter). While the expansion of IoT devices enables the development of attractive services, there are serious concerns about the increased risk of attacks using these devices to threaten the security and privacy of users [20]. For example, this is illustrated by the Mirai botnet that has compromised up to 600,000 devices [2].

Traditional security solutions, such as firewalls or network intrusion detection systems are generally based on the monitoring and analysis of network communications over a central device, such as a gateway or a proxy, in order to block unauthorized communications or to raise alarms. However, these solutions are generally limited in scope. In particular:

- Direct communications between devices using ad-hoc or proprietary protocols are generally not covered, although they are commonly used.
- Only a few protocols are supported (mainly WiFi, though other bands, such as 400–500 MHz or 800–900 MHz, are commonly used in the IoT), and existing security monitoring solutions are not efficient to cover simultaneously the wide range of protocols used in smart spaces.
- State-of-the-art solutions are not easy to adapt to take into account new devices or novel communication protocols.

New techniques are clearly needed to complement existing solutions and fill these major gaps. In this paper, we present a novel approach for the detection of intrusions in IoT environments that aims to achieve this goal. In particular, our objective is to provide an *Intrusion Detection System* (IDS), called RIDS (for *Radio Intrusion Detection System*), that satisfies the following properties:

(P1) Protocol-agnostic: it does not rely on the protocol specification and is, therefore, able to adapt to proprietary protocols and future wireless communication threats and technologies,
(P2) Multi-Band monitoring capabilities: it can be easily configured to monitor multiple (not necessarily contiguous) frequency bands simultaneously,
(P3) Non-invasive: it does not require any instrumentation or modification of the monitored devices and their functionalities,
(P4) Diagnosis support: it provides relevant information to assist in the analysis of detected anomalies and the identification of compromised devices.

The first three properties are fulfilled by monitoring radio communication activities on predefined frequency bands using *Software Defined Radio* (SDR) technology. Indeed, such an approach is generic and does not rely on prior knowledge of wireless protocols specification or modulation technology used by IoT devices. IDS based on radio activities have rarely been explored (see discussion in Related Work, Section 2).

RIDS is based on the assumption that attacks are likely to affect the observed radio activity either by using a frequency band (i.e., a wireless technology) never used by users or with unusual behavior or intensity. We propose to use machine learning techniques to model the observed radio communications, and then detect anomalies when there are significant deviations from the regular activities recorded during the training phase.

The radio activity data needed for the training of the machine learning models and the detection of anomalies is collected over large predefined frequency bands based on the deployment of a set of probes using SDR technology. The probes are scattered in the monitored space to cover a large area. Diverse hardware technologies (using different antennas) are used to maximize the sensitivity of the entire system. Multiple frequency bands can be covered simultaneously.

We use deep learning techniques to process the large amount of data collected during the experiments. More specifically, we rely on autoencoders [13], to characterize the usual radio activity patterns. These models have proven to be efficient in various application areas [9, 37], including anomaly detection [7, 11, 28, 33]. They can adapt to potential changes in the environment without requiring additional information on user behavior or attacks. Besides detecting anomalies,

RIDS is designed to provide users with a diagnosis of the anomaly (property **P4**) along three dimensions: time (start and end date of the anomaly), frequency (main radio frequency around which the anomaly is observed), and space (geographic coordinates of the position from which the anomalous activity was initiated).

Such diagnosis information is useful to help human operators and *Security Operations Centers* (SOC) identify the device causing the anomaly and those affected by the anomaly. This information could also be used by other protection systems, such as IoT Sentinel [22], to isolate compromised devices. It also allows one to investigate the objectives of the attacker, which could be i) to modify devices to prepare for an intrusion [15]; ii) to collect confidential information from them [23]; iii) to control them to bounce elsewhere [31] or to modify their functionality [30].

To assess our system, we deployed RIDS in an operational environment with about 20 IoT devices. We generated malicious activities representative of real-world IoT attacks (such as deauthentication, rogue access point, TV spoofing, and De Bruijn injection) on various protocols (including WiFi, Zigbee, BLE, ESB) and frequency bands (400–500 MHz, 800–900 MHz and 2.4–2.5 GHz) that are used by a wide range of connected devices.

We summarize our major contributions in this paper as follows:

- We present RIDS, an intrusion detection system based on autoencoder neural networks. It supports anomaly detection and diagnosis in heterogeneous IoT environments using radio communication activities recorded via multiple probes. The main advantages of RIDS compared to existing solutions are its independence from the specifications of wireless protocols used in the monitored environment, as well as its ability to monitor multiple, non-contiguous frequency bands, its non-intrusiveness, low cost and ease of use, and its ability to detect and diagnose the time, frequency and location of anomalies. To our knowledge, there is no equivalent IDS in the state of the art that offers all these features.
- We performed various experiments, with and without generated attacks, to evaluate the efficiency of RIDS in the frequency bands 400-500 MHz, 800-900 MHz, and 2.4-2.5 GHz. These experiments show that anomalies over 400-500 MHz and 800-900 MHz frequency bands are generally well detected: their occurrence time is diagnosed with high precision (93.0% on average) and high recall (generally higher than 90%). These results illustrate that RIDS can provide the users or administrators of smart environments with relevant information to identify and investigate attacks. On the other hand, low precisions and recalls are observed in the 2.4-2.5 GHz frequency band. Possible reasons for these low performances are discussed in the paper. Nevertheless, this frequency band is already well covered by other specific intrusion detection mechanisms.
- We collected a new dataset containing about 160 GB of radio activity over three broad frequency bands in an operational environment. Besides, this dataset includes real-world attacks performed with an attack framework. All data are available on request by contacting the authors. To our knowledge, there is no similar public dataset designed to validate IDS for IoT environments (see Section 2).

The paper is organized as follows. Section 2 first discusses the related work on IoT network intrusion systems and highlights our main novel contributions. Section 3 presents an overview of the threat model and the architecture of RIDS. Data acquisition is discussed in Section 4 and Section 5 deals with radio activity modeling and anomaly detection. Section 6 describes how temporal, frequency, and spatial diagnoses are achieved. Section 7 describes the experiments carried out. Their results are analyzed in Section 8. Section 9 discusses some of the limitations and threats to validity of our experiments. Finally, Section 10 concludes the paper and discusses future work.

2 RELATED WORK

IoT security has been the subject of several studies in recent years. In this section, we focus on attacks on wireless protocols and on solutions to mitigate these attacks at different protocol layers.

2.1 IoT attacks

Private, public, and professional environments are increasingly connected [1]. Recent wireless communication protocols commonly used in such environments, such as Bluetooth, BLE or protocols that use open frequency bands (e.g. 868 MHz or 433 MHz bands in Europe) have been targeted by new classes of attack. S. Jasek [16] describes several attacks on BLE and a tool named *Gattacker* to intercept, replay or inject messages between two devices. C. W. Badenhop et al. [4] analyzed the security of the Z-Wave routing protocol. They discovered several vulnerabilities that allow an outsider to modify routes between nodes. E. Ronen et al. [31] proposed a proof-of-concept of a Zigbee worm that exploits weaknesses within well-known smart-bulbs. D. Cauquil [8] proposes a similar solution called *Btlejuice* to perform Man-in-the-Middle attacks on BLE devices. M. Newlin [23] developed *MouseJack*, a tool able to sniff or inject keystrokes on wireless, non-Bluetooth mice and keyboards. Y. Bachy et al. [3] discovered vulnerabilities on DVB-T communications used for smart TVs that allowed them to inject crafted messages and remotely take control of the smart TVs. A. Francillon et al. [12] used relay attacks to remotely open cars that use *Passive Keyless Entry and Start* (PKES) technology on 315 MHz and 433 MHz bands.

As several attacks (e.g. Mirai [2]) rely on the exploitation of vulnerabilities in common Internet protocols, traditional countermeasures are generally investigated to cope with such attacks. For instance, N. Dhanjani [10] reports several weaknesses in the application layers of IoT devices that can be easily detected by known security measures on a gateway. Some other detection and mitigation solutions, more specifically developed for IoT environments, have also been proposed in recent research work, as presented in the next section.

2.2 Detection and mitigation solutions

Several solutions have been developed to detect and mitigate IoT threats. A recent detailed survey on intrusion detection and prevention techniques for wireless communication networks is provided in [18]. Most of the reported solutions are dedicated to a particular protocol or provide only high-level information on how the proposed techniques are implemented. In the following, we discuss two categories of existing solutions according to the highest protocol layers in which detection or mitigation is performed. The first category includes detection and protection techniques that are carried out at the link or higher layers. These techniques require demodulation of the communication signal and are not generic. The second category focuses on the physical layer and radio activities only, without demodulation. A comparative analysis of these techniques considering the properties **P1** to **P4** presented in the previous section is also provided.

2.2.1 Link, network, transport or application layers. Sung [35] has investigated the possibility of detecting curious adversaries using probes that monitor radio activity based on the Received Signal Strength Indicator (RSSI) provided by the Bluetooth Low Energy technology. This work shows advanced results on the detection of abnormal behavior in a smart environment. However, it focuses only on physical intrusions targeting BLE protocol.

Meidan et al. [21] used a machine learning classifier to identify the set of whitelisted devices based on their TCP/IP communications. They can detect unauthorized devices by identifying deviations in their IP communications compared to the model of whitelisted communications. This solution focuses only on TCP communications. Local objects that directly communicate via ad-hoc protocols without going through a central gateway cannot be monitored.

Siby et al. [34] designed IoTScanner to monitor heterogeneous protocols. IoTScanner relies on dedicated hardware for each monitored protocol (e.g. a Bluetooth dongle, etc.). Hence, it cannot be easily maintained: each time a new protocol is used in the environment, a new dedicated hardware device must be designed and integrated into IoTScanner. Moreover, the development of dedicated hardware for proprietary or non-standardized protocols can be time-consuming and costly.

Raza et al. [29] developed SVELTE, an IDS for 6LoWPAN and IPv6 networks, which requires the modification of devices to set up a secure connection between them to detect routing attacks such as sinkhole and forwarding attacks. A hybrid approach with both a centralized and distributed solution is proposed: their IDS is deployed on the devices and on the 6LoWPAN border router. One of the main limitations of SVELTE is the need to modify the IoT devices, which makes it impractical in a real connected environment with multiple objects from different manufacturers.

The IoT Sentinel approach developed by Miettinen et al. [22] mainly aims at identifying and isolating vulnerable devices. It is designed to 1) accurately identify the types of devices connected to an IoT network by modeling and fingerprinting WiFi communications, 2) identify each device vulnerabilities by matching its type to the *Common Vulnerabilities and Exposures* (CVE) database, and 3) isolate vulnerable devices to prevent them from being attacked by configuring the main firewall. This solution focuses solely on WiFi communications, and permanently disables vulnerable objects if they cannot be patched, making them useless. Besides, zero-day attacks cannot be prevented because the isolation strategy is based on the CVE database.

2.2.2 Physical layer and radio solutions. New anomaly detection techniques based on the monitoring of radio-communication activities have been recently explored [26, 28, 32]. Such methods have the advantage of being independent of the specification of IoT communication protocols.

The solution investigated by O'Shea and al. [26] is based on recurrent neural networks. It only addresses the detection of radio communication anomalies without providing means to help the user or an expert to understand and analyze the anomalies. The detection performance of their approach is evaluated by applying synthetic mutations to selected samples of the recorded radio signals. Finally, all their samples are limited to a small bandwidth (about 20 MHz).

Recently, S. Rajendran and al. [28] presented SAIFE, an unsupervised anomaly detector. It uses an adversarial autoencoder to output interpretable features about the electromagnetic spectrum. They focus on long-range wireless protocols in smart cities (LoRa, Sigfox, etc.). Their work aims to provide tools for managing overloaded frequency bands. Promising results have been obtained on the analysis of anomalies on different datasets, synthetic and real, using SDR. This detector is also able to localize the frequency of an anomaly. However, there are no details on how this is achieved. This approach has some similarities with our solution. Nevertheless, 1) this paper deals with ultra narrow band protocols, whose communication profiles are quite different from the protocols of smart environments composed of different connected objects, 2) very few details are provided about experiments, making the comparison with our work quite difficult, and 3) they do not address the spatial diagnosis of the detected anomalies.

It is also noteworthy that this research as well as the research carried out by O'Shea and al. [26], do not specifically target the detection of malicious and security related anomalies.

In our previous work presented in [32], we proposed a first solution, named RadIoT, that was designed to cover IoT related attacks and to monitor radio activities over a wide spectrum of frequencies. The solution is protocol-agnostic and independent of protocol layers specifications. It monitors the radio communication activities generated by the connected objects, without demodulation, and profiles these activities using machine learning techniques, in particular autoencoders. The approach is designed to support intrusion detection in smart homes. Several experiments were carried out in a controlled environment in which one single probe was deployed to monitor radio

communications and various attacks targeting a heterogeneous set of IoT devices were generated and detected with high accuracy. However, the considered experimental conditions are not representative of more complex operational environments and large smart spaces, in which multiple probes need to be deployed. Also, this system was not designed to help the users troubleshoot the anomaly, which is particularly detrimental since a wide spectrum of the radio activity is monitored, making it cumbersome to find the origin of an anomaly and diagnose it. In fact, a skilled user would need to manually check the radio activity measurements at the time of the attack to know which protocol is affected and whether there is a false alarm or not.

2.2.3 Summary of existing solutions and our contributions. To highlight the main contributions of this paper, Table 1 outlines how each IDS discussed in the related work section, fulfills the four properties (**P1, P2, P3**, and **P4**) presented in the introduction. The frequency bands covered by the different solutions are also indicated. RIDS is the only solution that is designed and implemented to satisfy all these properties. It can be noticed also that existing security solutions mainly cover communication protocols operating in the 2.4–2.5 GHz frequency band. Attacks using other frequency bands such as 400–500 MHz or 800–900 MHz are generally not covered. The contributions presented in this paper are intended to fill this gap.

Table 1. Solutions and their properties

Solution	**P1** Protocol agnostic	**P2** Band	**P3** Non invasive	**P4** Diagnosis		
				Temporal	Frequential	Spatial
IoT Sentinel		NA	✓	✓		
SVELTE		NA		✓		
BLE RSSI		2.4–2.5 GHz	✓	✓		
Whitelist detection		2.4–2.5 GHz	✓	✓		
IoT Scanner	✓	2.4–2.5 GHz	✓	✓	✓	
O'Shea	✓	80–100 MHz, 850–860 MHz 1.93–1.94 GHz, 2.40–2.42 GHz	✓	✓		
SAIFE	✓	✓	✓	✓	✓	
RadIoT	✓	✓	✓	✓		
RIDS	✓	✓	✓	✓	✓	✓

3 THREAT MODEL AND OVERVIEW

3.1 Threat model

In a smart context, the attacker can proceed remotely (e.g. without physically interacting with the devices) either from the Internet (or more generally outside the smart environment), or within the smart environment. Its objectives may be to: i) modify the devices to prepare a home intrusion [15], ii) collect confidential information within the devices [23], iii) control devices in order to bounce elsewhere [31], or iv) control devices to affect their functionality [30].

The attack will result in malicious interactions with the devices of the smart environment, that may occur concurrently with legitimate interactions. To compromise the targeted devices, the attacker can either use his own devices or use an already compromised one:

(1) from a location that is never used by users;
(2) using a wireless technology never used by users;
(3) during an unusual period of the day;
(4) or with unusual intensity.

These malicious activities are likely to have a strong impact on radio-activities. Hence, in our study, we focus on the monitoring of the radio-activities generated by such interactions, considering both the time when the radio-activities are observed and the corresponding radio-frequency.

3.2 Overview

The design of *RIDS* relies on three main components, as shown in Figure 1 :

- *Data acquisition*: that is based on the deployment of several radio probes to monitor and process the radio-activities;
- *Model learning*: which is dedicated to the elaboration during the training phase of an autoencoder model that reflects legitimate radio-activities
- *Detection and diagnostic*: which processes radio activities collected during the detection (also called operation) phase and raises an alarm when an anomaly is detected and provides information about its time, radio-frequency and spatial location.

The *radio probes* are intended to monitor the radio activities of the environment, without any modification and without any prior information about the protocols used within the environment (see property **P1**). Moreover, they must be easily configurable to monitor multiple frequency bands simultaneously (property **P2**). All observations are processed by the probes to generate *spectrograms* as shown in figure 2. The deployment of the probes must allow an optimized coverage of the considered environment and rely on a security expert able to identify the best locations to place the probes. Since these probes do not require any modification of the devices or the environment, RIDS satisfies the property **P3**.

To distinguish legitimate radio activities from malicious ones, a model of legitimate communications has to be defined. Therefore, two main phases are necessary, as shown in Figure 1. First, the *training* phase aims at learning the model of legitimate radio activities (*Model learning*). Then, during the *detection* phase, new radio activities are compared to this model to detect deviations from the model, i.e., anomalies. At the end of this phase, each anomaly is analyzed to retrieve diagnosis information about its time, radio-frequency and spatial location (later called *temporal, frequential and spatial diagnosis*), which satisfies the property **P4**.

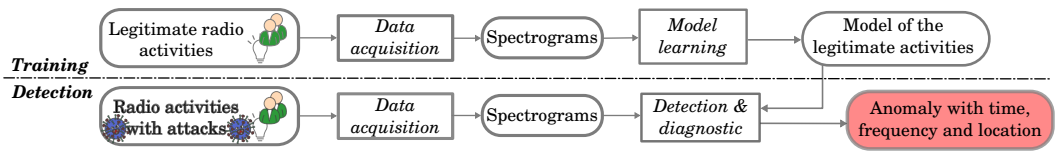

Fig. 1. RIDS overview

4 DATA ACQUISITION

Data acquisition is achieved using several radio probes positioned within the smart space. The following requirements are put forward for the selection of these probes:

- Capability to monitor activities on different frequency bands;
- Low cost;
- Easy to configure with the help of a security expert;
- Portable;
- Non-invasive, i.e., does not require any modification of IoT devices.

Radio probes are composed of two main components: a Software-Defined Radio (SDR) device and a chip that processes signal data and produces a spectrogram.

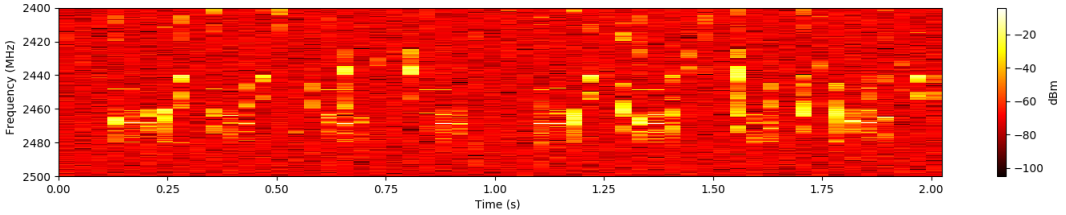

Fig. 2. Spectrogram example: two seconds of data over 2.4–2.5 GHz

The SDR device allows the monitoring of radio signals over a wide frequency band, unlike dedicated hardware components that directly monitor the signal received on a particular frequency. They can also use software processing to decode and recover the original information sent (packets, commands, etc.). However, this software processing must be configured, by specifying e.g., the type of demodulation. As our goal is to monitor multiple protocols with a single probe, without any assumption about the physical layer, we did not use this technique and directly analyzed the electromagnetic radio signals.

Many SDR receivers implement a *sweep* mechanism allowing the monitoring of wide frequency bands (up to 6 GHz) with a single device. A description of this mechanism can be found in [14]. In this case, the signal data cannot be reconstructed but the power received on each frequency can be measured. The spectral power of each frequency is estimated using several Fast-Fourier-Transform (FFT) over the frequency band (hence the name *sweep*). Another advantage is the ability to monitor disjoint frequency bands (e.g., 400-500 MHz, 800-900 MHz and 2.4-2.5 GHz.) and to change these bands with minimal reconfiguration. On the other hand, the time resolution must be adjusted to provide optimal tradeoffs and reduce the overhead associated with measuring each sweep. Many low-cost SDR devices that directly implement the sweep mechanism are available, such as the HackRF One [1].

The second component of the probe processes the outputs of the sweeps performed by the SDR device and continuously generates their spectrograms. One spectrogram $S_{p,b}$ is generated per continuous frequency band b and per probe p. Therefore, $S_{p,b}(t, f)$ is the radio activity power measured by the probe p on the band b, at time t and frequency f. As an example, Figure 2 shows two seconds of a spectrogram covering 2.4–2.5 GHz band. Each column represents one sweep in this band.

5 RADIO-ACTIVITY MODELING AND ANOMALY DETECTION

The goal is to build a model based on machine learning that accurately reflects the legitimate spectrograms and to compute a time series of the errors that correspond to the deviations between this model and the measurements. The error should be high in case of an anomaly, low otherwise.

Among the wide variety of state-of-the-art machine learning models, we have chosen the autoencoder neural network that has proved to provide good results in anomaly detection [7, 28, 33].

The process for the elaboration of the autoencoder model is summarized in Figure 3. The different steps are explained in the following sub-sections. All the notations are defined in the appendix A.

5.1 Autoencoder detection model

An autoencoder is a deep neural network [24] that is trained to produce outputs that are identical to its inputs. It is structured as a non-recurrent feedforward neural network with the same input and output size. It has an intermediate bottleneck layer that allows learning a compressed representation

[1] https://greatscottgadgets.com/hackrf/

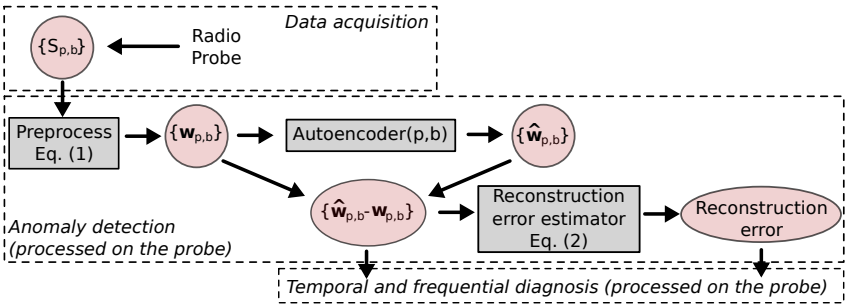

Fig. 3. Anomaly detection diagram

of the inputs. The purpose of compression is to reduce the overfitting of the autoencoder. As the probes produce an uninterrupted stream of measurements, we have to slice them into chunks that can be fed to the autoencoder. To detect the presence of an anomaly in a spectrogram chunk, the learned autoencoder is used to generate the reconstructed chunk. If the original spectrogram chunk is "typical" and contains no anomaly, the reconstruction error should be small since the autoencoder has been learned by minimizing this error. Otherwise, the reconstruction error should be higher. Instead of using a single model, we have chosen to learn a model for each spectrum band we monitor and for each probe. This choice is motivated by the assumption that anomalies can be detected by analyzing each frequency band separately. Using several models instead of just one provides several advantages: each model requires less data for its training and new frequency bands and probes can be added without having to relearn the other models. The main motivation for choosing the autoencoder is that it can provide a reconstruction error per input feature, which is very useful to locate the radio frequency associated with an anomaly (Section 6.2). Besides, autoencoders can be learned from a large amount of data (for example, in our experimental setup, about 2GB of data are measured each day).

The architecture of the proposed model is composed of four layers corresponding to 1) the inputs, 2) one 1D temporal convolutional layer, 3) one bottleneck dense layer and, 4) an output dense layer. The 1D convolutional layer is used to take advantage of the time-invariance of the spectrograms. Indeed, the sliding time windows used to slice the spectrogram could be shifted and still have the same information. The details of the implementation are presented in Section 7. An example of this reconstruction is illustrated in Figure 5.

5.2 Data preprocessing: noise removal and scaling

A very common loss function for autoencoder learning is the root mean square error (RMSE) between the input $\mathbf{x}$ and the output $\widehat{\mathbf{x}}$, expressed as follows:

$$RMSE(\mathbf{x}, \widehat{\mathbf{x}}) = \sqrt{\frac{1}{|\mathbf{x}|}\sum_{i=1}^{|\mathbf{x}|}(\mathbf{x}(i) - \widehat{\mathbf{x}}(i))^2}$$

However, we cannot directly use this loss function because of the high proportion of noise in the spectrograms. Let's consider the example of a spectrogram chunk S that monitors 100 frequencies such as there is a signal on only one frequency, the rest being noise. This proportion is not absurd, since we are monitoring wide frequency bands. The noise values vary around a mean value. As noise is randomly distributed, the autoencoder cannot estimate it better than this mean value. Suppose that the mean absolute error of the noise reconstruction is 5 dB. Let's assume the autoencoder does

not reconstruct the signal correctly, resulting in a mean absolute error of 30 dB in that frequency. The root mean squared error is therefore $\sqrt{\frac{1}{100}30^2 + \frac{99}{100}5^2} \approx \sqrt{9+25} \approx 5.83$.

In this example, most of the reconstruction error can be attributed to the noise. It can mislead the detection phase that uses the accumulated reconstruction error to identify anomalies. This phenomenon is amplified by the fact that the signal may concern only a small fraction of the monitored frequencies. To solve this problem and improve the training of our model, we remove the reconstruction error due to noise by flattening the noise level to a single value. More precisely, measurements below a lower bound B_l are mapped to 0, those above an upper bound B_u are mapped to 1 and the rest are scaled linearly between 0 and 1. This scaling in the $[0; 1]$ interval is adapted to the learning algorithm of the autoencoder.

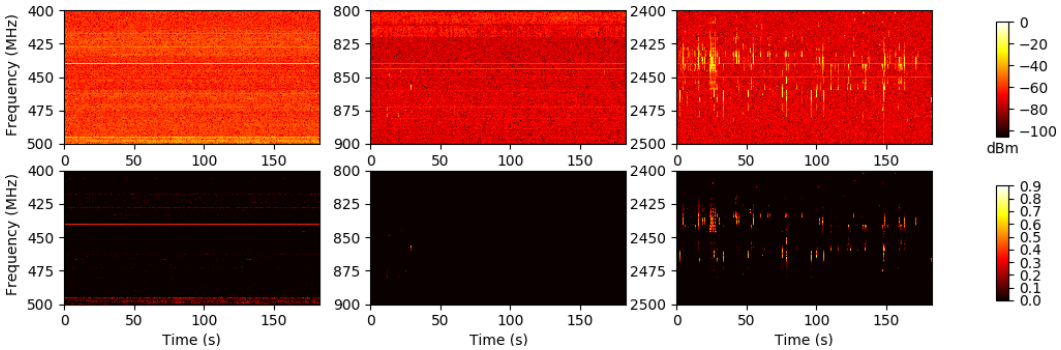

Fig. 4. A spectrogram chunk (top) and its preprocessed version (bottom)

More formally, we define[2] $\mathbf{w}_{p,b}$ the processed spectrogram of $S_{p,b}$ and its corresponding loss function:

$$\mathbf{w}(t,f) = \begin{cases} 0 \text{ if } S(t,f) < B_l \\ \frac{S(t,f)-B_l}{B_u-B_l} \text{ if } B_l \leq S(t,f) \leq B_u \\ 1 \text{ if } S(t,f) > B_u \end{cases} \qquad (1) \qquad RMSE(\mathbf{w}, \widehat{\mathbf{w}}) = \sqrt{\frac{1}{|\mathbf{w}|}\sum_t \sum_f (\mathbf{w}(t,f) - \widehat{\mathbf{w}}(t,f))^2}$$

An example is presented in Figure 4. At the top, there are three unprocessed spectrograms (one for each frequency band). At the bottom, their processed counterparts, after noise removal.

5.3 Model learning and reconstruction error

The models are trained by minimizing the root mean squared error of the reconstructed input with the "adam" optimizer [19]. A sliding time window is applied to the spectrograms to generate the inputs. The error estimation is based on the difference between the spectrogram chunk and its reconstruction.

In Figure 5, a chunk is represented on the left, its reconstruction in the middle and their absolute difference on the right. The yellow dotted horizontal line is an attack. We can see that it is not reconstructed in the middle figure and that it is therefore present in the difference. Even though the autoencoder is learned by minimizing the root mean squared error of the reconstructed input, we did not use this metric to produce the reconstruction error processed by the anomaly detection for the same reason explained in Section 5.2: since the anomalies generally affect a limited part of

[2]For the sake of clarity, we will thereafter omit the p and b subscripts and write $\mathbf{w}$ and S.

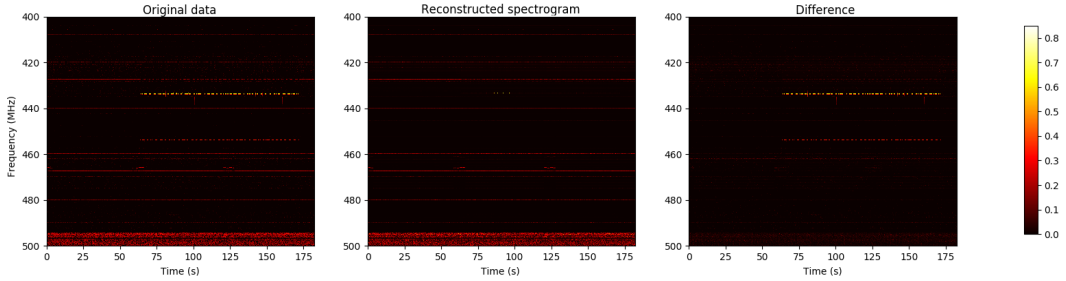

Fig. 5. A spectrogram chunk, its reconstruction and their absolute difference. The yellow dotted horizontal line is a "de Bruijn" injection [17] and the red dotted horinzontal line is one of its harmonic.

the spectrum we are monitoring, their large reconstruction error can be diluted in the small but ubiquitous reconstruction error of the normal traffic. To limit this effect, we use the 4-norm distance: $d_4(a,b) = \sqrt[4]{(a-b)^4}$ to ensure that the reconstruction error penalizes much more significant errors than small errors. We adapt the RMSE to this distance and finally define our reconstruction error as:

$$RE(\mathbf{w}, \widehat{\mathbf{w}}) = \sqrt[4]{\frac{1}{|\mathbf{w}|}\sum_t \sum_f (\mathbf{w}(t,f) - \widehat{\mathbf{w}}(t,f))^4} \tag{2}$$

Note that we could have used this as a loss function for the learning phase, but we think this would significantly penalize the small errors and affect the convergence of the algorithm.

For each input, a score is produced with the reconstruction error defined in equation (2). To get a better time resolution of the reconstruction error, a sliding time window is applied to the spectrograms to generate the inputs. This reconstruction error is then processed by the anomaly time diagnosis process.

6 TEMPORAL, FREQUENTIAL AND SPATIAL DIAGNOSIS

The purpose of this section is to estimate the occurrence time of the attack as well as the radio frequency associated with the detected anomaly. For instance, this information can be used to identify the attacker (by correlating the time of the attack and the logs of an access control system) or to identify the type of objects targeted (though the radio frequency associated with the attack).

Figure 6 summarizes the main steps of the temporal and frequential diagnosis processes.

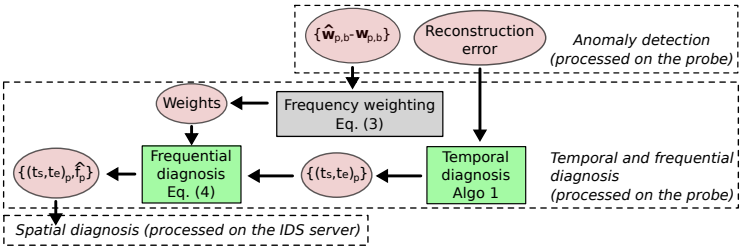

Fig. 6. Time and frequency diagnosis

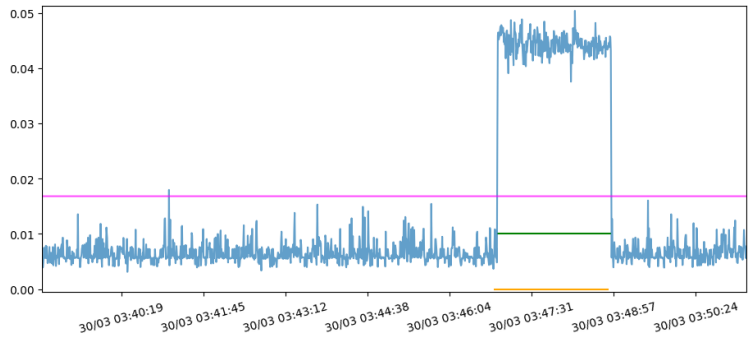

Fig. 7. A reconstruction error time series (in blue), the threshold T_1 (in pink), the period of a two minutes DoS attack on 433 MHz (in orange) and the detected period (in green)

6.1 Temporal diagnosis

To detect an anomaly, the temporal diagnosis component analyzes the time series of the reconstruction error and checks whether it crosses a threshold $T_1(t)$. This threshold should be learned from the training set since it depends on the radio activity of the environment. An example of a reconstruction time series, the interval of the attack (a two minute DoS attack on 433 MHz) and the detected interval is presented in Figure 7.

Two difficulties may arise in choosing this threshold. First, the environment may not be completely controlled, which means that the training set may contain some anomalies. Setting the threshold to the highest reconstruction error of the training set may not be optimal. A more robust threshold would be to estimate it with a high percentile rather than the maximum value. Second, the reconstruction error is correlated with traffic activity, which varies over time. For this reason, this threshold is defined as a function of time $T_1(t)$ (for example, a lower threshold may be set at night, since there is generally less activity at that time).

Algorithm 1: The temporal diagnosis algorithm

Input: e, the RE error function (blocks if no data yet) for probe p on band b,
T_1 an error threshold that depends on the time t,
T_2, a threshold on cumulative error

Algorithm `TemporalDiagnosis(`e, T_1, T_2`)`

```
1  t ← FirstDate(e); D_b(p) ← ∅
2  loop                                                    // process forever
3      while e(t) < T_1(t) do t ← t + 1                    // get first crossing
4      cumulative_error ← 0; detection ← False; t_s ← t
5      while e(t) ≥ T_1(t) do          // accumulate while still above the threshold
6          cumulative_error ← cumulative_error + |e(t) − T_1(t)|
7          if ¬detection and cumulative_error ≥ T_2 then
8              detection ← True; AdvertiseAlarm()          // advertise as soon as possible
9          t ← t + 1
10     if detection then t_e ← t − 1; D_b(p) ← D_b(p) ∪ {(t_s, t_e)}    // memorize the new interval
```

The temporal diagnosis algorithm (Algorithm 1) estimates the start (t_s) and the end (t_e) dates of an anomaly. The start date is the time when the error exceeds the threshold; similarly, the end date

of the anomaly is the time when the reconstruction error falls below the threshold $T_1(t)$ (or, more precisely, the time *just before* the threshold is crossed again). However, this method could generate too many false positives. For example, as shown in Figure 7, the reconstruction error may randomly cross the threshold $T_1(t)$. A more resilient detector can be defined by considering the cumulative error, which is the sum of the errors above $T_1(t)$. When the reconstruction error falls below $T_1(t)$, the cumulative error is compared to another threshold (called the cumulative threshold, T_2). If this cumulative threshold T_2 is exceeded, the detection is confirmed. Otherwise, the detection of that anomaly is dropped. The cumulative error allows the system to detect long and subtle anomalies as well as short and intense anomalies. For each probe p, it computes $D_b(p)$, the union of all the detection intervals on the frequency band b.

6.2 Frequential diagnosis

The diagnosis of the frequency of the anomalous radio activity is based on the evaluation of the 4-norm distance between each point of the reconstructed spectrogram $\hat{\mathbf{w}}$ and the measured spectrogram $\mathbf{w}$ between the start date t_s and the end date t_e estimated by the anomaly time diagnosis component. A weight is associated with each frequency f, as stated in (3). The estimated frequency associated with the detected anomaly is the frequency with the maximum weight, as shown in (4).

$$weight(f) = \sum_{t=t_s}^{t_e} d_4(\mathbf{w}(t,f), \hat{\mathbf{w}}(t,f)) \qquad (3) \qquad\qquad \hat{f} = \underset{f}{\operatorname{argmax}}\, weight(f) \qquad (4)$$

This estimator is mainly intended for single-frequency anomalies (such as an attack on an 868 MHz device), or anomalies involving multiple and close frequencies (such as the failure of a WiFi router on a single channel). Another approach could use the weighted percentile to generate a frequency band of the anomaly (by computing, for example, the frequency band that contains 90% of the weights). We also believe that a more sophisticated frequency diagnosis could produce a list of frequencies of interest by applying clustering techniques to the list of weights defined above. However, such a system is out of the scope of this article.

6.3 Spatial diagnosis

As discussed in Section 2, one of the main contributions of our work is to provide a comprehensive diagnosis assistant. Besides estimating the time and the frequency associated with the detected anomalies, the spatial location of the device that generated the communications related to the detected anomaly is also an important piece of information to provide. This diagnostic is based on the combination of data from different probes.

Some approaches aimed at performing spatial diagnosis of radio signals without modifying the target device (called device-free approaches) have been studied by [36] and [35] in the context of intrusion detection for civil environments or elderly care. This work used link layer data such as the *Received Signal Strength Indicator* (RSSI) to estimate the location of users within an environment. It is noteworthy that most of the existing effective techniques are based on RSSI [25]. However, the RSSI information is provided by only a few protocols within the link-layer of the receiver. For protocols that do not comply with the 802.11 standard, other ways should be considered to provide this information, or other alternatives should be considered for spatial diagnosis implementation. A possible solution could be to use the k-nearest neighbors algorithm, as first proposed by [5]. However, this algorithm requires a calibration phase which is generally not easy to implement. For this reason, we decided to use the weighted centroid algorithm proposed by [6] to estimate the location of an anomaly from the power measured by multiple probes, without any prior calibration.

As illustrated in the diagram of Figure 8, our diagnostic of the spatial location of the anomaly can be decomposed into three steps:

- Inference of the median time and communication frequency associated with the anomaly from the fusion of the time and frequency diagnosis information provided by each probe (*Fusion*)
- Estimation for each probe of the received power strength around the estimated median time and frequency associated with the anomaly (*Power estimation*)
- Estimation of the geographic position corresponding to the estimated power (*Spatial diagnosis*).

These three steps are described in the following.

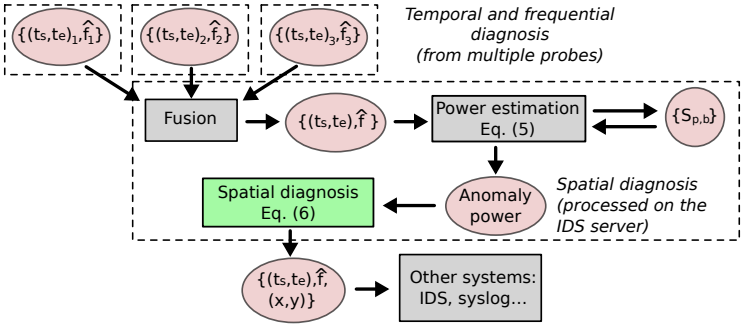

Fig. 8. The spatial diagnosis is based on the temporal and frequential diagnosis of each probe

An anomaly is considered to exist when it is detected by at least one probe. Other decision strategies based e.g. on the detection by the majority of probes or by all probes could also be studied.

More formally, the merged anomaly time D_b is the union of $D_b(p)$: $D_b = \bigcup_p D_b(p)$. The fusion of the frequency diagnosis information is performed afterward; it is the mean of the frequencies associated with the anomaly according to each probe that detected the anomaly. The merged diagnosis results are then sent to the *power estimation* component.

As presented in Figure 8, the timestamps and frequency of the anomalies are used to request the corresponding (unprocessed) spectrograms $\{S_{p,b}\}$ from each probe. Then, if we denote (t_s, t_e) the timestamp of the anomaly and $\hat{f}$ the frequency, the median power of the probe p on band b (that contains $\hat{f}$) is estimated by extracting a narrow band of δ_f MHz around $\hat{f}$ between t_s and t_e, as shown in equation (5).

$$MedianPower(p, t_s, t_e, \hat{f}) = \text{median}\{S_{p,b}(t, f) \mid t \in [t_s, t_e], f \in [\hat{f} - \frac{\delta_f}{2}, \hat{f} + \frac{\delta_f}{2}]\} \tag{5}$$

The last step of our approach consists in using the weighted centroid algorithm proposed by [6] to estimate the location of an anomaly from the power measured by multiple probes.

Given the position (x_p, y_p) of the probe p and the median power $MedianPower(p, t_s, t_e, \hat{f})$ of the anomaly measured by the probe p, the location (x_a, y_a) of the anomaly is estimated by:

$$\underset{(x_a, y_a)}{\text{argmin}} \sum_p \left(MedianPower(p, t_s, t_e, \hat{f}) - s_{min} \right)^{\lambda} d((x_p, y_p), (x_a, y_a))^2 \tag{6}$$

where d is the Euclidean distance, s_{min} is a detection threshold strength and λ is a free parameter.

7 EXPERIMENTS

To assess the efficiency of RIDS, we have set up an experimental smart environment, that is as realistic as possible in terms of wireless technologies and user behaviors, as well as potential attacks

and anomalies. The autoencoder model implemented in RIDS was built after a training period without attacks. The detection is assessed with an environment composed of legitimate users and generated attacks.

We conducted two experiments. In the first experiment, the efficiency of the IDS is assessed by processing off-line the generated radio activity spectrograms. The probes and the IDS server processing ran on a server with an Intel Core i7-7700 CPU at 4.2 GHz, 32 GB of RAM, and an Nvidia graphics card GTX 1080 Ti. This experiment evaluates the ability of our system to detect anomalies and to provide a precise diagnostic. The second experiment assesses on-line the real-time performance of a low-cost probe.

7.1 Experimental setup

The experimental environment is illustrated in Figure 9 and corresponds to a server room hosting master's degree students during their training period. It is a crowded and regularly visited room used for coffee breaks. During the experiment, the room was equipped with additional wireless devices: they are described in Table 2 with their purpose and their wireless technologies. The behavior of the users is neither simulated nor fully controlled, as they can interact with connected objects and bring their own devices (smartphones, laptops, smartwatches, etc.). Also, some actions were automated to promote wireless activity and to ensure that each object is used (for example, the doorbell randomly rang twice a day).

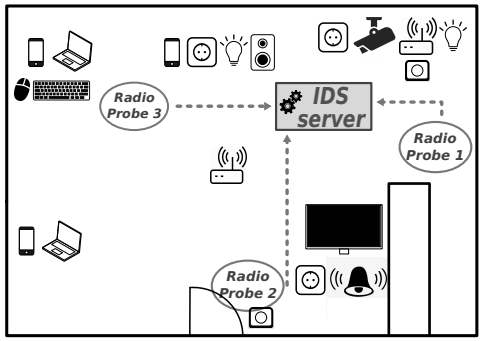

Fig. 9. Smart environment

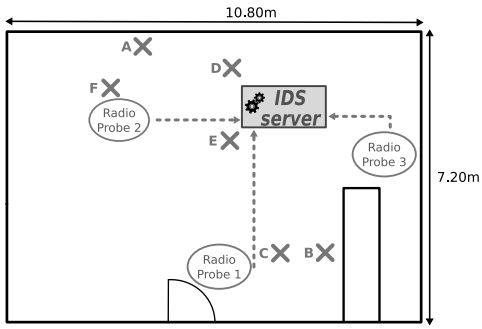

Fig. 10. Attacks location in the environment

Table 2. Connected objects of the smart experimentation room

Name	Aim	Wireless technology	Qty
D-Link Camera	Live Streaming Video	WiFi/2.4–2.5 GHz	1
Phones, Laptops	General Internet Services	WiFi/2.4–2.5 GHz	5+
Philips Hue	Lightbulb	ZigBee/2.4–2.5 GHz	1
Beewi Smartbulb	Lightbulb	BLE/2.4–2.5 GHz	1
Keyboard & mouse	Wireless peripherals	ESB/2.4–2.5 GHz	2
WiFi Access Points	Wireless connectivity	WiFi/2.4–2.5 GHz	2
Smart outlet Nityam	Wireless electrical outlet	868 MHz	3
Bricelec doorbell	Wireless door bell	433 MHz	1
HackRF One	Automatic emitter	433 MHz	1
HackRF One	Automatic emitter	868 MHz	1
Samsung Smart TV	Television	470–800 MHz & 2.4–2.5 GHz	1

Three probes (numbered 1, 2, and 3) were deployed. Each one is composed of a HackRF One as the SDR peripheral and a Raspberry Pi to process the outputs. Each Raspberry Pi is connected via Ethernet to the IDS server. For the ease of the experimentation, we decided to log the spectrograms inside the IDS server. We monitored three bands: 400–500 MHz and 800–900 MHz (generally used for home automation and small sensor networks), and 2.4–2.5 GHz (WiFi, Zigbee, etc.)[1]. Each probe has an antenna that is particularly sensitive to one band: 400–500 MHz for probe 1, 800–900 MHz for probe 2, and 2.4–2.5 GHz for probe 3. We set the frequency resolution of the sweeps to 10 measurements per MHz, which gives 3000 measures for the three 100 MHz bands. Each sweep takes about 37.5 ms to scan all frequency bands, which corresponds to a time resolution of about 27 sweeps per second. From these numbers, we can estimate the data throughput of our system. Each second, each probe measures about 27 sweeps composed of 3000 points. Each measurement is stored in a one-byte format, to limit the size of the data. Therefore, 80 kB of data are measured per probe every second. In total, more than 2 GB of data are processed every day.

As summarized in Figure 11 in Appendix B, the anomaly detection and the temporal and frequential diagnoses are processed on each probe. The IDS server only takes care of the spatial diagnosis based on the outputs of the probes.

7.2 Model parameters

7.2.1 Preprocess. In our experiment, the noise level is about -65 dBm and the probes never measured a power level above 0 dBm, so we set the upper bound B_u and lower bound B_l to 0 dBm and -50 dBm respectively.

7.2.2 Error estimator. The spectrograms are split into slices whose dimensions are 16×1000: 16 for the time dimension (about 600 ms) and 1000 for the frequency dimension (i.e. one spectrum band). The 600 ms duration is a compromise: if it's too low, the autoencoder is not able to learn a long radio pattern; if it's too large, the number of parameters of the model to learn is too high and the learning phase needs too many data. The hyperparameters values chosen for the autoencoder are as follows:

(1) the inputs, composed of 16×1000 features;
(2) a 1D temporal convolutional layer with 500 filters and a temporal window size of 5, with rectified linear (ReLu) activation;
(3) the bottleneck layer, a dense layer with 2000 features, with sigmoid activation;
(4) an dense layer with 16×1000 outputs, with sigmoid activation.

The same architecture is used for all bands, so the results are not affected by the specific behavior of each band. In fact, the optimal size of the bottleneck layer (the third layer) probably depends on the traffic in the frequency band. Bands with low traffic (such as 800–900 MHz) would require a thinner bottleneck layer to avoid overfitting; on the contrary, higher traffic bands (such as 2.4–2.5 GHz) would require a larger bottleneck layer to properly learn its behavior. Therefore, the value of 2000 features is a tradeoff. This model has been implemented and learned with Keras and TensorFlow back-end. A more extensive hyperparameter search would certainly enhance our results, but such work is out of the scope of this article.

During the learning phase, the overlap of the consecutive sliding windows is 80% to maximize the amount of data available for training. During the testing set, it is set to 20%.

7.2.3 Temporal diagnosis. Two values were used for the error threshold (T_1): one for weekday between 8 a.m. and 7:30 p.m. and one for weekend and night. We estimated this error threshold T_1 with the 99^{th} percentile of the score time series of the training set for each band and each probe.

We also estimated the T_2 parameter (cumulative error threshold) as the 99.995^{th} percentile of the cumulative errors on the error function for all probes and all bands on the training set.

7.2.4 Spatial diagnosis. The detection threshold power is set to $s_{min} = -65$ dBm (the noise level of the room) and the free parameter of the model to $\lambda = 2$. The width of the frequency band around the central frequency is set to $\delta_f = 0.2$ MHz.

7.3 Attacks

7.3.1 Generated attacks. To generate anomalies that are as close as possible to the malicious activities that can be performed in a smart environment, we studied a wide range of attack classes targeting connected objects and IoT environments. Ronen and al. [31] presented a Zigbee attack on a previous version of the lightbulb Philips Hue. They exploited a previous leak of the master key to emit a high-powered Zigbee signal to the lightbulb to take control of it from outside a building. Y. Bachy and al. [3] presented an attack on smart TV aiming at spoofing the legitimate transmitter of a DVB-T channel and transmitting malicious data streams (such as faked video or even some URL referencing malicious code). For that purpose, they had to crush the legitimate signal by transmitting with a significantly stronger power. S. Kamkar [17] presented some replay attacks on garage doors and vehicles, and also developed OpenSesame[3], a device that uses De Bruijn sequences to open wirelessly the same kind of objects.

Table 3. Classes of injected attacks and unexpected anomalies (in orange)

ID	Name	Technology	Type	Freq./Band	#Injection
			400–500 MHz		
1	scan433-17	433 MHz	Scan 433 17 dBm	433 MHz	256
2	DoS433-27	433 MHz	DoS 27 dBm	433 MHz	50
3	DoS433-40	433 MHz	DoS 40 dBm	433 MHz	13
4	TV-spoofing	DVB-T	TV spoofing [3]	485–499 MHz	12
5	bruijn	433 MHz	De Bruijn injection [17]	433 MHz	13
6	probfail-1	Radio Probe	Probe failure	400–500 MHz	×
7	anomaly462	Monitor	Unexpected monitor signal	462 MHz	×
8	anomaly467	Monitor	Unexpected monitor signal	467 MHz	×
			800–900 MHz		
9	scan868	868 MHz	Scan 868 20 dBm	868 MHz	279
10	DoS868	868 MHz	DoS 35 dBm	868 MHz	80
11	harmo433	866 MHz	Harmonics of ID 3 and 5	866 MHz	29
12	probfail-2	Radio Probe	Probe failure	800–900 MHz	×
			2.4–2.5 GHz		
13	blescan	BLE	BLE Scan	2.4–2.5 GHz	177
14	zigbeescan	Zigbee	Zigbee Scan	2.4–2.5 GHz	25
15	deauth	WiFi	Deauthentication	2.451–2.473 GHz	33
16	rogueAP	WiFi	RogueAP	2.461–2.483 GHz	83
17	esbinject	ESB	Injection ESB	2.4–2.5 GHz	66
18	injectzigbee	Zigbee	Injection Zigbee [31]	2.48 GHz	33

We used the open-source framework Mirage[4] to generate a large set of attacks on different technologies from multiple locations (labeled A to F, see Figure 10). Table 3 describes the different

[3]https://github.com/samyk/opensesame
[4]https://redmine.laas.fr/projects/mirage

classes of attacks that were performed during this experiment. Each attack class has been carried out for 2 minutes multiple times at different locations. In total, about 1100 attacks were generated.

7.3.2 Unexpected anomalies. In addition to our injected attacks, five unexpected anomalies occurred during the experiment. "probfail-1" and "probfail-2" correspond to unexpected failures of the probe. During these anomalies, the probe did not halt but only returned noise with no signal on 400–500 MHz and 800–900 MHz bands. We could have removed this period from the dataset, but we decided to keep it since it is a legitimate anomaly that should be detected. Anomaly "harmo433" on 866 MHz is an unexpected harmonic of the attacks "DoS433-40" and "bruijn" on 433 MHz. Its transmission power is not strong. Both "anomaly462" and "anomaly467" correspond to radio activities that do not match any device within our experimental environment (neither a malicious device nor a legitimate one). Our investigation shows that it has been produced by a laptop monitor brought by an intern. We did not expect a monitor to generate such a strong signal.

7.4 Experimental protocol

We collected the training set from March 19th to March 27th, 2019 (9 days). We collected two testing sets. The attacks of the first dataset (collected from March 28th to April 3rd, 2019) were issued from position F (see Figure 10). The attacks of the second dataset (collected on May 7th, 8th, 13th and 14th, 2019) were issued from different locations (A to E). For each dataset, the same set of objects presented in Table 2 is used. As it is a real-case scenario, we also collected the activities of some unknown devices, brought and removed by users. All data are available on request by contacting the authors.

7.4.1 Detection evaluation. To evaluate the detection capabilities of RIDS, we measure the proportions of attacks with a non-empty intersection with the set of the detection intervals estimated by the probes on the same band. In other words, an attack is deemed detected if an alarm is raised on the right band during its duration.

7.4.2 Temporal diagnosis evaluation. Evaluation of the effectiveness of temporal diagnosis is based on a variety of classical classification metrics based on four metrics: true positive TP (when there is an anomaly and the system raises an alarm), false positive FP (no anomaly but an alarm is nonetheless raised), true negative TN (no anomaly and no alarm), and false negative FN (there is an anomaly but no alarm is raised). The metrics we use are:

- Precision = $\frac{TP}{TP+FP}$: rate of relevant detection among all alerts.
- Recall = $\frac{TP}{TP+FN}$: rate of anomalies that have been detected.
- False positive rate: FPR = $\frac{FP}{FP+TN}$: rate of false positives in the absence of anomalies.
- Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$: rate of correct predictions (be it an alarm or an absence of alarm).

To assess anomaly time diagnosis effectiveness, we extend to intervals the usual discrete concepts of precision and recall defined by [27]. Let us denote A_b the union of the attack time intervals on band b, $A_{b,i}$ the union of the time intervals associated with attack ID i on band b and D_b, the union of time intervals derived from the time location detection algorithm applied to band b (defined in Section 6.3). We define the continuous true positive, false positive, true negative, and false negative as (where ^{c} is the complement):

$$TP_{b,p} = |A_b \cap D_{b,p}| \qquad FP_{b,p} = |A_b^c \cap D_{b,p}| \qquad TN_{b,p} = |A_b^c \cap D_{b,p}^c| \qquad FN_{b,p} = |A_b \cap D_{b,p}^c|$$

Besides, we define the precision per band P_b and the recall of attack ID i (visible on the band $b(i)$), denoted R_i, as:

$$P_b = \frac{|A_b \cap D_b|}{|D_b|} \quad (7) \qquad R_i = \frac{|A_{b(i),i} \cap D_{b(i)}|}{|A_{b(i),i}|} \quad (8)$$

Note that it is not possible to measure the precision per attack as the type of attack is not an output of the system.

7.4.3 Frequential diagnosis evaluation. To assess the performance of the frequency diagnosis component, we focus on the estimated frequencies corresponding to an actual attack. More exactly, a detection interval $[t_s, t_e]$ is considered valid if there exists an attack $[a_s, a_e]$ on the same band such that, if we denote $[i_s, i_e] = [t_s, t_e] \cap [a_s, a_e]$, the two following conditions are met: $\frac{i_e - i_s}{t_e - t_s} \geq 0.5$ (at least half of the detection covers the attack) and $\frac{i_e - i_s}{a_e - a_s} \geq 0.5$ (at least half of the attack is detected).

7.4.4 Spatial diagnosis evaluation. Similarly to frequency diagnosis assessment, we only evaluate the estimated location corresponding to an actual attack (using the same criteria) with a correctly estimated frequency (with an error less than 1 MHz). The evaluated metric is the median of the Euclidean distance between the actual location of the source of the attack and the estimated location.

8 RESULTS

The proportions of detected anomalies (i.e. anomalies during which an alarm was raised) are presented in Table 4. Let us note that only the attacks with a known number of occurrences are included. Probe 2 detected most of the attacks. It is probably due to its location in the room: as shown in Figure 10, as it is close to several attack locations.

Three groups of attack classes can be identified:

- very well detected attack classes (all attacks on 400–500 MHz and 800–900 MHz except "scan868" and "harmo433"): at least 90% of attacks are detected;
- partially detected attack classes ("scan868" and "harmo433"): 76% of "scan868" attacks and 45% of "harmo433" attacks are detected;
- undetected attack classes on 2.4–2.5 GHz: less than 1% of attacks are detected.

The low detection results on the 2.4–2.5 GHz band are expected. The following reasons can be given to explain this lack of detection:

- Some attack classes cannot be detected with a demodulation-free detector based on physical layer information. These attacks include WiFi deauthentication, rogue AP, and ESB injection: the radio traffic generated by these attacks is very similar to legitimate radio activity.
- The time resolution of the SDR probes is too low to detect some types of radio activities (as explained in Section 4). For example, BLE communications are not measured.
- Besides, radio activity in the 2.4–2.5 GHz band is far more difficult to learn (see Figure 4), which generates a higher noise level in the reconstruction error.

To summarize, these results illustrate the effectiveness of RIDS in detecting attacks on the 400–500 Mhz and 800–900 Mhz bands that are not well covered by traditional techniques, while these techniques provide a better solution for the 2.4–2.5 GHz frequency band. In the following, we will only focus on the results related to 400–500 Mhz and 800–900 Mhz bands.

8.1 Temporal diagnosis evaluation

The recall per attack of the time diagnosis is summarized in Table 5. The recall is very close to the proportion of detected anomalies (Table 4), which means that the detection time intervals encompass almost all of the detected attacks. As a reminder, the precision and the recall are computed from the detection and the attack intervals, as explained in Section 7.4.2 with Equations (7) and (8).

The precision is also relatively high: between 79.79% and 99.13% for 400–500 MHz and between 96.38% and 97.76% for 800–900 MHz. As regards the recall performance, the effectiveness varies

Table 4. Proportion of detected anomalies

Name	Probe 1	Probe 2	Probe 3
400–500 MHz			
scan433-17	52.59%	97.61%	0%
DoS433-27	100%	100%	4.00%
DoS433-40	100%	100%	100%
TV-spoofing	100%	100%	100%
bruijn	92.31%	100%	100%
800–900 MHz			
scan868	65.82%	76.00%	65.09%
DoS868	83.75%	95.00%	65%
harmo433	44.83%	44.83%	3.45%
2.4–2.5 GHz			
ID 13–18	$\leq 1\%$	$\leq 1\%$	$\leq 1\%$

Table 5. Temporal diagnosis recall

Name	Probe 1	Probe 2	Probe 3
400–500 MHz			
scan433-17	51.52%	96.63%	0%
DoS433-27	99.19%	99.21%	4.00%
DoS433-40	99.19%	99.87%	95.77%
TV-spoofing	99.20%	99.85%	99.00%
bruijn	77.88%	96.60%	84.48%
probfail-1	99.83%	99.88%	9.59%
anomaly462	3.51%	100%	5.36%
anomaly467	70.75%	29.70%	6.18%
800–900 MHz			
scan868	64.97%	74.13%	64.56%
DoS868	82.09%	93.33%	63.95%
harmo433	40.02%	40.73%	3.35%
probfail-2	0%	0%	3.32%

Table 6. Temporal diagnosis metrics per band and per probe

Metrics	400–500 MHz			800–900 MHz		
	Probe 1	Probe 2	Probe 3	Probe 1	Probe 2	Probe 3
Precision	99.13%	79.79%	87.54%	97.76%	96.38%	97.40%
Recall	21.67%	98.59%	6.69%	57.65%	65.39%	52.18%
FPR	0.06%	7.46%	0.28%	0.10%	0.18%	0.10%
Accuracy	81.94%	93.93%	78.32%	96.99%	97.44%	96.61%

significantly depending on the probes for two reasons: the probes do not have the same antenna and they are not at the same distance from the place where the attacks were issued. For example, probe 2 (recall: 98.59% on 400–500 MHz and 65.39% on 800–900 MHz) is close to attacking locations A, D, E, and F while probe 3 (recall: 6.69% on 400–500 MHz and 52.18% on 800-900 MHz) is partially obstructed by some furniture in the room.

The false positive rate (FPR) is very low (below 0.3%) except for probe 2 on 400–500 MHz. False positives observed for this probe concern almost exclusively the TV band (485–499 MHz) that has a high intensity and is difficult to learn as it seems random. Only probe 2 is affected for two reasons: first, its antenna is particularly sensitive to the 400-500 MHz band and second, it is close to the place where the TV signal comes from.

Finally, the accuracy is high for all the probes except for probe 1 and probe 3 on 400–500 MHz: these results can be explained by their low recall.

8.2 Frequential diagnosis evaluation

The results of the frequential diagnosis evaluation are presented in Table 7. As explained in Section 7.4.3, we are only interested in attacks correctly detected by the time diagnosis component. The cells with $\times$ correspond to attacks that are not detected (cf. Table 5). They cannot, therefore, be evaluated.

With only a few exceptions, the frequency estimation error does not exceed 0.1 MHz. This highly precise estimation should provide valuable support to an administrator to further investigate the type of device at the origin of the detected anomaly. Besides, the estimated frequency can be reliably used to extract the median received power of the attack (cf. Section 6.3). The less accurate

estimation is observed on probe 3 for "DoS433-27" attack. Note that the recall measured for this type of attack on this probe is also very low.

Table 7. Median frequency error

Name	Frequency	Probe 1	Probe 2	Probe 3
400–500 MHz				
scan433-17	433 MHz	0.1 MHz	0.1 MHz	×
DoS433-27	433 MHz	0.1 MHz	0.1 MHz	63.4 MHz
DoS433-40	433 MHz	0.1 MHz	0 MHz	0 MHz
TV-spoofing	485–499 MHz	0 MHz	0 MHz	0 MHz
bruijn	433.8 MHz	0 MHz	0 MHz	0 MHz
anomaly462	462 MHz	×	0.1 MHz	×
anomaly467	467 MHz	0 MHz	0 MHz	×
800–900 MHz				
scan868	868 MHz	0.1 MHz	0.1 MHz	0.1 MHz
DoS868	868 MHz	0.1 MHz	0.1 MHz	0.2 MHz
harmo433	867.7 MHz	0 MHz	0 MHz	0 MHz

8.3 Spatial diagnosis evaluation

The results of the spatial diagnosis are summarized in Table 8. Each cell indicates the median Euclidean distance for each detected attack. The locations where the attacks are launched are displayed in Figure 10.

Table 8. Median Euclidean distance for each attack class on each location

Name	Pos. A	Pos. B	Pos. C	Pos. D	Pos. E	Pos. F	Mean
scan433-17	-	-	-	-	-	3.63 m	3.63 m
DoS433-27	-	-	-	-	-	3.96 m	3.96 m
DoS433-40	MA	0.82 m	0.51 m	MA	1.24 m	-	0.86 m
TV-spoofing	MA	0.45 m	0.35 m	MA	1.92 m	-	0.91 m
bruijn	MA	0.95 m	0.32 m	-	2.12 m	MA	1.13 m
scan868	2.45 m	×	-	×	×	2.91 m	2.68 m
DoS868	3.16 m	2.47 m	2.67 m	3.13 m	1.15 m	2.19 m	2.46 m
harmo433	2.13 m	×	-	2.40 m	-	-	2.27 m
Mean	2.68 m	1.17 m	0.91 m	2.77 m	1.61 m	3.17 m	1.95 m

In the Table, "-" indicates that the attack was not performed from the corresponding position. For example, attacks "scan433-17" and "DoS433-27" were only carried out from position F, and attack "scan868" was not carried out from position C. "×" represents attacks that were not detected, and therefore could not be located. "MA", for *Multiple Anomalies*, refer to unexpected anomalies "anomaly462" and "anomaly467" that occurred at the same time as the attack identified in the corresponding line of the table. Indeed, our frequential diagnosis algorithm (cf. Section 6.2) only estimates a single main frequency. Since this frequency is used to estimate the median received power, and the algorithm cannot be applied in case of simultaneous attacks.

Overall, the results are rather stable (with a standard deviation of 1.09 m). Whenever an attack is made, our model can predict the location of the emitter with a mean accuracy of 1.95 m. Considering the size of the room and the information provided by the time and frequency diagnoses, a security administrator should be able to quickly identify the malicious or faulty device.

Note that the predicted locations of "scan433-17" and "DoS433-27" are generally less precise than the others. It can be explained by the fact that the "DoS433-27" attack was tested from only one single location and that the "scan433-17" attack was more difficult to detect because of its lower power intensity, making it difficult for the model to locate it precisely.

In our experiment, the detection was done offline, which requires storing all the data of the test set. The precision of each point was reduced to a single byte to limit the amount of stored data. That led to a resolution of 1 dBm. In the case of an online system, this restriction is useless (e.g., one could only keep the last hour of data). It means that the median received power could have a better resolution. In that case, we would expect the spatial diagnosis to be more accurate.

We believe these results could be enhanced with a better probe setup. In this experiment, the three probes had different antennas with different sensitivities (cf. Section 7.1). It has probably affected the accuracy of the location estimation algorithm as the received power depends not only on the distance but also on the sensitivity of the antenna.

8.4 Detection time complexity

The most expensive step is the reconstruction of $\mathbf{w}$ (cf. Figure 5) that can be computed in $O(N_w)$ operations, where N_w is the number of weights of the feedforward neural network. This number depends on the hyper-parameters of the network, i.e. on the number of layers and the type and the size of each layer. The other operations (preprocessing, reconstruction error computation, temporal diagnosis, frequency weighting, frequential diagnosis, fusion, power estimation, and spatial diagnosis) are in $O(|\mathbf{w}| \times |A|)$, where $|\mathbf{w}|$ is the size of one spectrogram chunk and $|A|$ is the duration of an attack.

8.5 Real-time capability

The results presented in the previous sub-sections are obtained by processing the spectrograms off-line. To assess the usability of our approach in on-line conditions with low-cost probes, we carried out a second set of experiments with a Raspberry Pi 4B (with 4 GB of RAM) with Ubuntu Server 64bits 19.10.1. The Raspberry Pi processed on-line the spectrograms received on the three frequency bands for about one hour ($\sim$ 6000 spectrogram chunks). The model computations were performed with the CPU of the Raspberry Pi (and not its GPU) with Tensorflow. The Raspberry Pi has four cores: one core was dedicated to the model computations while the other cores handled the HackRF One data stream.

We collected several metrics presented: the load of the processing core, the load of the HackRF One cores, the memory usage of the IDS processes, the temperature, and the power consumption (including the Raspberry Pi, a fan and the HackRF One). Their average values remained constant during the experiment. The fan kept the temperature low: 37.8 °C / 100 °F. The overall power consumption is also low: 4.5 W. Finally, this probe can process on-line the spectrograms while using 39.80% of the computational power of its core and 1998 MB of RAM. Hence, this low-cost probe is more powerful than necessary for real-time analysis.

9 DISCUSSION AND THREATS TO VALIDITY

The experiment was carried out in a complex working environment with realistic operation conditions, including different commercially available objects using heterogeneous communication protocols. Only a small set of behaviors were automated, but all were based on real scenarios. The attacks are also based on or inspired by real scenarios and were injected with the Mirage tool, a penetration testing framework for IoT networks. It is interesting to note that unexpected anomalies have also been correctly detected, which confirms the relevance of our approach. So far, however, only elementary attacks were injected. In the future, some attack scenarios combining multiple

attacks on different bands should be implemented to evaluate the effectiveness of our detection and diagnosis approach. Moreover, more automated usage scenarios of some connected devices could be tested as the IoT concept underlines less human interaction with deployed devices. Additionally, very short attacks may not be detected because the temporal resolution of the measurements in our experiment is one sweep every 37.5 ms. However, we think this weakness is only technology-dependent and should be mitigated by the use of more expensive hardware (such as a USRP with the SweepSense modification [14]).

As for the cost of the approach, it is affordable as the main component is the probe that costs only about 300 $. If the models are already learned, a low-powered computer can easily perform detection in real-time.

Another important issue we have not addressed is the optimization of probe placement. In our opinion, this choice should be made by a security expert in the same way as for the placement of cameras in the deployment of an alarm system against physical intrusion.

Spatial diagnosis may not be acceptable in every environment and may raise concerns about the privacy of the users. Our solution can be deployed with temporal and frequential diagnoses only.

The experiments we have carried out are a first proof of concept, aimed at demonstrating the relevance of our approach. The results for anomaly detection and diagnosis, in the temporal, frequency, and space dimensions are very promising, especially for the communication bands that are not well covered by traditional intrusion detection solutions. We still need to improve the spatial diagnosis accuracy, even though our preliminary results are encouraging as they can provide valuable information for a security expert to identify the origin of an attack in complex and dynamic IoT environments. It should be noted that this solution is not intended to be used alone, but rather in addition to other solutions that can detect attacks on higher communication layers (e.g. Mirai [2]), especially on well-known protocols that operate in the 2.4-2.5 GHz band. Our approach could be combined with these solutions in a SOC, to improve the detection coverage.

10 CONCLUSION AND PERSPECTIVE

In this paper, we proposed a novel anomaly detection approach for IoT environments, based on the analysis and the profiling of radio communications, without any signal demodulation. This approach implements a threefold diagnosis: temporal, frequential, and spatial. The first experiments carried out to assess the relevance of the approach yielded fairly good and encouraging results. We were able to detect all the anomalies except one (“probfail-2”) on 400-500 MHz and 800-900 MHz bands and were able to diagnose them. Such a detection approach seems to be an efficient complement to traditional approaches that are instead based on the analysis of communication flows after signal demodulation. Indeed, RIDS can detect anomalies on frequency bands that are not usually monitored or for which there are non-specific probes, as they correspond to proprietary protocols.

We can assess our system regarding the properties and objectives stated in the introduction. By design, and as our experiments can attest, our demodulation-free IDS is generic and is capable of adapting to different wireless technologies. It is also scalable: one only needs to add probes to cover a larger area. RIDS has been successful in a complex, real-world working environment with a rather large number of users. It can used with low-cost equipment for the real-time detection and diagnosis of anomalies. Finally, it does not need to be calibrated by running attacks.

This anomaly detection solution would be best used as a first detection layer. Its ability to produce temporal, frequential, and spatial diagnosis of the anomaly with minimal hypothesis could help specialized tools to refine their detection, such as IoT Sentinel [22]. The estimated frequency associated with the detected anomaly should be very useful to identify the protocol of the attack. Its spatial location should make the identification of the origin of the attack easier.

The promising results obtained so far need to be confirmed by carrying out other experiments likely to improve the accuracy of our results. Regarding the spatial diagnosis, for instance, we still have to perform more attacks from many different locations, both inside and outside the experimental environment, and to analyze to what extent the results of these experiments may provide valuable insights to improve our detection algorithms. We also plan to carry out experiments with more powerful probes, such as SDR probes with a higher temporal or frequential resolution, which should improve the accuracy of our detection. To cope with the low detection capabilities of our approach on the 2.4-2.5 GHz frequency band, we could improve our results by splitting the 2.4-2.5 GHz into short slices and establishing a model of the legitimate activities dedicated to these slices. Such an approach was adopted in [32] and provided promising results.

A long-term perspective to this research work concerns the characterization of the detected anomaly, in terms of attack types. In other words, is it possible, only from the temporal, frequency, and spatial diagnosis of the anomaly, to deduce the type of attack that is currently ongoing? For instance, a DoS attack probably has a different radio profile than an attack aimed at advertising a fake device. Such an approach could lead us to further explore the limitations and advantages of demodulation-free and protocol-agnostic solutions.

REFERENCES

[1] Ala I. Al-Fuqaha, Mohsen Guizani, Mehdi Mohammadi, Mohammed Aledhari, and Moussa Ayyash. 2015. Internet of Things: A Survey on Enabling Technologies, Protocols, and Applications. *IEEE Communications Surveys and Tutorials* 17, 4 (June 2015), 2347–2376. https://doi.org/10.1109/COMST.2015.2444095

[2] Manos Antonakakis, Tim April, Michael Bailey, Matt Bernhard, Elie Bursztein, Jaime Cochran, Zakir Durumeric, J. Alex Halderman, Luca Invernizzi, Michalis Kallitsis, Deepak Kumar, Chaz Lever, Zane Ma, Joshua Mason, Damian Menscher, Chad Seaman, Nick Sullivan, Kurt Thomas, and Yi Zhou. 2017. Understanding the Mirai Botnet. In *26th USENIX Security Symposium*. USENIX Association, 1093–1110.

[3] Yann Bachy, Vincent Nicomette, Mohamed Kaâniche, and Eric Alata. 2019. Smart-TV security: risk analysis and experiments on Smart-TV communication channels. *Journal of Computer Virology and Hacking Techniques* 15, 1 (March 2019), 61–76. https://doi.org/10.1007/s11416-018-0320-3

[4] Christopher W. Badenhop, Scott R. Graham, Benjamin W. Ramsey, Barry E. Mullins, and Logan O. Mailloux. 2017. The Z-Wave routing protocol and its security implications. *Computers & Security* 68, 4 (April 2017), 112–129. https://doi.org/10.1016/j.cose.2017.04.004

[5] Paramvir Bahl and Venkata N. Padmanabhan. 2000. RADAR: An In-Building RF-Based User Location and Tracking System. In *Proceedings IEEE INFOCOM 2000, The Conference on Computer Communications*. IEEE, 775–784. https://doi.org/10.1109/INFCOM.2000.832252

[6] Jan Blumenthal, Ralf Grossmann, Frank Golatowski, and Dirk Timmermann. 2007. Weighted centroid localization in zigbee-based sensor networks. In *2007 IEEE International Symposium on Intelligent Signal Processing (ISISP'07)*. IEEE, 1–6. https://doi.org/10.1109/WISP.2007.4447528

[7] Andrea Borghesi, Andrea Bartolini, Michele Lombardi, Michela Milano, and Luca Benini. 2019. Anomaly detection using autoencoders in high performance computing systems. In *Proceedings of the AAAI Conference on Artificial Intelligence*. AAAI, 9428–9433. https://doi.org/10.1609/aaai.v33i01.33019428

[8] D. Cauquil. 2016. *Btlejuice: The bluetooth smart mitm framework*. Technical Report.

[9] Jan Chorowski, Ron J. Weiss, Samy Bengio, and Aäron van den Oord. 2019. Unsupervised speech representation learning using wavenet autoencoders. *IEEE/ACM Trans. Audio, Speech and Lang. Proc.* 27, 12 (Dec. 2019), 2041–2053. https://doi.org/10.1109/TASLP.2019.2938863

[10] Nitesh Dhanjani. 2015. *Abusing the internet of things: blackouts, freakouts, and stakeouts*. O'Reilly Media, Inc.

[11] Qingsong Feng, Yabin Zhang, Chao Li, Zheng Dou, and Jin Wang. 2017. Anomaly detection of spectrum in wireless communication via deep auto-encoders. *The Journal of Supercomputing* 73, 7 (July 2017), 3161–3178. https://doi.org/10.1007/s11227-017-2017-7

[12] Aurélien Francillon, Boris Danev, and Srdjan Capkun. 2011. Relay Attacks on Passive Keyless Entry and Start Systems in Modern Cars. In *Proceedings of the Network and Distributed System Security Symposium (NDSS 2011)*. NDSS Symposium, 1–15.

[13] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep learning*. MIT press.

[14] Yeswanth Guddeti, Raghav Subbaraman, Moein Khazraee, Aaron Schulman, and Dinesh Bharadia. 2019. SweepSense: Sensing 5 GHz in 5 Milliseconds with Low-cost Radios. In *16th USENIX Symposium on Networked Systems Design and*

Implementation, NSDI 2019, Boston, MA, February 26-28, 2019. USENIX Association, 317–330.

[15] Grant Ho, Derek Leung, Pratyush Mishra, Ashkan Hosseini, Dawn Song, and David Wagner. 2016. Smart Locks: Lessons for Securing Commodity Internet of Things Devices. In *Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security (ASIA CCS '16)*. Association for Computing Machinery, 461–472. https://doi.org/10.1145/2897845.2897886

[16] Slawomir Jasek. 2016. *Gattacking Bluetooth Smart Devices*. Technical Report.

[17] Samy Kamkar. 2015. DEFCON 23 : Drive it like you hacked it: new attacks and tools to wirelessly steal cars.

[18] Abril Khan, Amjad Mehmood, Shafiullah Khan, Muhammad A. Khan, Zeeshan Iqbal, and Wali K. Mashwani. 2020. A survey on intrusion detection and prevention in wireless ad-hoc networks. *Journal of Systems Architecture* 105, 1 (May 2020), 101701. https://doi.org/10.1016/j.sysarc.2019.101701

[19] Diederik P. Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. (2015). arXiv:http://arxiv.org/abs/1412.6980

[20] Constantinos Kolias, Angelos Stavrou, Jeffrey Voas, Irena Bojanova, and Richard Kuhn. 2016. Learning Internet-of-Things Security "Hands-On". *IEEE Security and Privacy* 14, 1, 37–46. https://doi.org/10.1109/MSP.2016.4

[21] Yair Meidan, Michael Bohadana, Asaf Shabtai, Martín Ochoa, Nils Ole Tippenhauer, Juan David Guarnizo, and Yuval Elovici. 2017. Detection of Unauthorized IoT Devices Using Machine Learning Techniques. (2017). arXiv:arXiv:1709.04647

[22] Markus Miettinen, Samuel Marchal, Ibbad Hafeez, N. Asokan, Ahmad-Reza Sadeghi, and Sasu Tarkoma. 2017. IoT SENTINEL: Automated Device-Type Identification for Security Enforcement in IoT. In *37th IEEE International Conference on Distributed Computing Systems (ICDCS'17)*. IEEE, 2177–2184. https://doi.org/10.1109/ICDCS.2017.283

[23] Marc Newlin. 2016. *MouseJack, KeySniffer and Beyond: Keystroke Sniffing and Injection Vulnerabilities in 2.4GHz Wireless Mice and Keyboards*. Technical Report.

[24] Andrew Ng et al. 2011. Sparse autoencoder. *CS294A Lecture notes* 72, 2011 (2011), 1–19.

[25] Henri Nurminen, Marzieh Dashti, and Robert Piché. 2017. A survey on wireless transmitter localization using signal strength measurements. *Wireless Communications and Mobile Computing* 2017, 1 (Feb. 2017), 1–13. https://doi.org/10.1155/2017/2569645

[26] Timothy J. O'Shea, T. Charles Clancy, and Robert W. McGwier. 2016. Recurrent Neural Radio Anomaly Detection. (2016). arXiv:arXiv:1611.00301

[27] James W Perry, Kent Allen, and Madeline M Berry. 1955. Machine literature searching x. machine language; factors underlying its design and development. *American Documentation (pre-1986)* 6, 4 (1955), 242. https://doi.org/10.1002/asi.5090060411

[28] Sreeraj Rajendran, Wannes Meert, Vincent Lenders, and Sofie Pollin. 2018. SAIFE: Unsupervised Wireless Spectrum Anomaly Detection with Interpretable Features. In *2018 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN'18)*. IEEE, 1–9. https://doi.org/10.1109/DySPAN.2018.8610471

[29] Shahid Raza, Linus Wallgren, and Thiemo Voigt. 2013. SVELTE: Real-time intrusion detection in the Internet of Things. *Ad Hoc Networks* 11, 8 (Nov. 2013), 2661–2674. https://doi.org/10.1016/j.adhoc.2013.04.014

[30] Eyal Ronen and Adi Shamir. 2016. Extended Functionality Attacks on IoT Devices: The Case of Smart Lights. In *IEEE European Symposium on Security and Privacy (EuroS&P)*. IEEE, 3–12. https://doi.org/10.1109/EuroSP.2016.13

[31] Eyal Ronen, Adi Shamir, Achi-Or Weingarten, and Colin O'Flynn. 2017. IoT Goes Nuclear: Creating a Zigbee Chain Reaction. In *2017 IEEE Symposium on Security & Privacy (SP'17)*. IEEE, 195–212. https://doi.org/10.1109/SP.2017.14

[32] Jonathan Roux, Eric Alata, Guillaume Auriol, Mohamed Kaâniche, Vincent Nicomette, and Romain Cayre. 2018. RadIoT: Radio Communications Intrusion Detection for IoT - A Protocol Independent Approach. In *17th IEEE International Symposium on Network Computing and Applications, NCA 2018, Cambridge, MA, USA, November 1-3, 2018*. IEEE, 210–218. https://doi.org/10.1109/NCA.2018.8548286

[33] Mayu Sakurada and Takehisa Yairi. 2014. Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis (MLSDA'14)*. Association for Computing Machinery, 4–11. https://doi.org/10.1145/2689746.2689747

[34] Sandra Siby, Rajib Ranjan Maiti, and Nils Ole Tippenhauer. 2017. IoTScanner: Detecting Privacy Threats in IoT Neighborhoods. In *Proceedings of the 3rd ACM International Workshop on IoT Privacy, Trust, and Security (IoTPTS '17)*. ACM, 23–30. https://doi.org/10.1145/3055245.3055253

[35] Yunsick Sung. 2016. Intelligent Security IT System for Detecting Intruders Based on Received Signal Strength Indicators. *Entropy* 18, 10 (Oct. 2016), 1–16. https://doi.org/10.3390/e18100366

[36] Moustafa Youssef, Matthew Mah, and Ashok Agrawala. 2007. Challenges: Device-Free Passive Localization for Wireless Environments. In *Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking (MobiCom '07)*. Association for Computing Machinery, 222–229. https://doi.org/10.1145/1287853.1287880

[37] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. 2019. Deep Learning Based Recommender System: A Survey and New Perspectives. *ACM Comput. Surv.* 52, 1, Article 5 (Feb. 2019), 38 pages. https://doi.org/10.1145/3285029

A NOTATIONS

Table 9. Parameters used within the article along with their experimental values

Parameter	Meaning	Experimentation value	Section
$\{b\}$	Frequency bands	400–500 MHz 800–900 MHz 2.4–2.5 GHz	4
(no notation)	Autoencoder input size	16×1000	5.1
B_l	Lower bound noise level	-50 dBm	5.2
B_u	Upper bound noise level	0 dBm	5.2
(no notation)	Training window overlap	80%	5.3
(no notation)	Testing window overlap	20%	5.3
δ_f	Anomaly bandwidth	0.2 MHz	6.3
s_{min}	Model free parameter	-65 dBm	6.3
λ	Model free parameter	2	6.3

Table 10. Notations used within the article

Notation	Meaning	Section
p	Probe index	4
$S_{p,b}$	Spectrogram	4
$\mathbf{w}$	Processed spectrogram	5.2
$RMSE(\mathbf{w}, \widehat{\mathbf{w}})$	Loss function for autoencoder learning	5.2
$d_4(a, b)$	4-norm distance	5.3
$RE(\mathbf{w}, \widehat{\mathbf{w}})$	Reconstruction error	5.3
e	Error time series	6.1
T_1	Detection threshold	6.1
T_2	Cumulative threshold	6.1
t_s	Start of an anomaly	6.1
t_e	End of an anomaly	6.1
$D_b(p)$	Union of detection intervals per probe p on band b	6.1
$weight(f)$	Weight of frequency f	6.2
$\hat{f}$	Estimated frequency of an anomaly	6.2
D_b	Union of detection intervals for all probes	6.3
d	Euclidean distance	6.3
$MedianPower(p, t_s, t_e, \hat{f})$	Median power measured by probe p	6.3
A_b	Union of attacks time intervals	7.4.2
$A_{b,i}$	Union of time intervals for attack i on band b	7.4.2
P_b	Precision per band b	7.4.2
R_i	Recall per band b	7.4.2

B FIGURE OF THE COMPLETE APPROACH

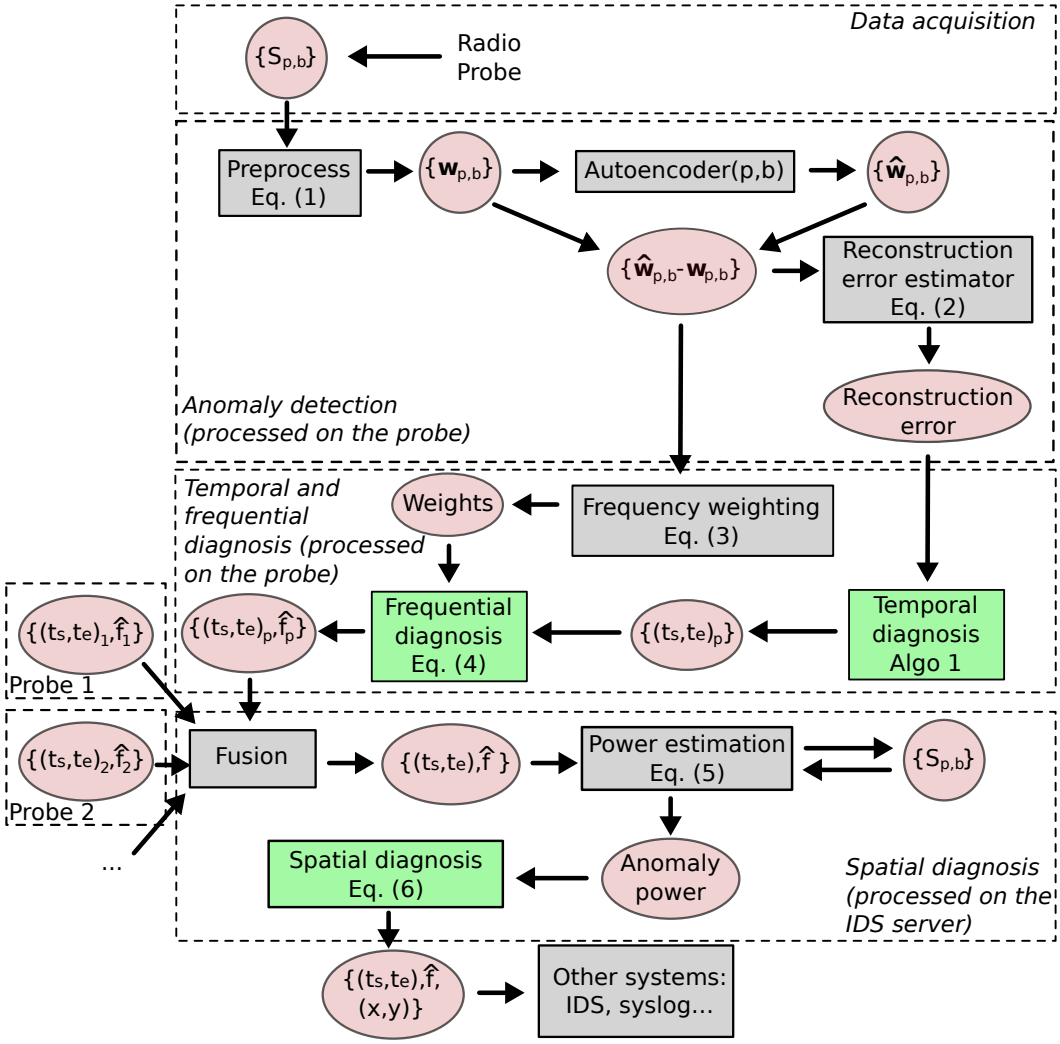

Fig. 11. The complete approach