Efficiently linking LoRaWAN identifiers through multi-domain fingerprinting

Samuel Pélissier[a], Abhishek Kumar Mishra[a], Mathieu Cunche[a], Vincent Roca[b], Didier Donsez[b]

[a]*University of Lyon, INSA-Lyon, Inria, CITI Lab, Lyon, France*
[b]*University Grenoble Alpes, Grenoble, France*

Abstract

LoRaWAN is a leading IoT technology worldwide, increasingly integrated into pervasive computing environments through a growing number of sensors in various industrial and consumer applications. Although its security vulnerabilities have been extensively explored in the recent literature, its ties to human activities warrant further privacy research. Existing device identification and activity inference attacks are only effective with a stable identifier. We find that the identifiers in LoRaWAN exhibit high variability, and more than half of the devices use them for less than a week. For the first time in the literature, we explore the feasibility of device fingerprinting in LoRaWAN, allowing long-term device linkage, i.e. associating various identifiers of the same device. We introduce a novel holistic fingerprint representation utilizing multiple domains, namely content, timing, and radio information, and present a machine learning-based solution for linking identifiers. Through a large-scale experimental evaluation based on real-world datasets containing up to 41 million messages, we study multiple scenarios, including an attacker with limited resources. We reach 0.98 linkage accuracy, underscoring the need for privacy-preserving measures. We showcase countermeasures including payload padding, random delays, and radio signal modulation, and conclude by assessing their impact on our fingerprinting solution.

Keywords: LoRaWAN, Sensors, IoT, Privacy, Security, Linkage attack, Multi-domain

1. Introduction

LoRaWAN stands out in the IoT ecosystem as a rapidly growing communication technology for sensor networks, which is already used by 350 million end-devices and connected to 181 operators around the world[1] The long range of transmission and low power consumption make this protocol a compelling candidate for a wide range of pervasive computing applications, from traffic monitoring to agriculture [1]. These applications, such as smart-cities or energy infrastructure [2–4], are highly sensitive with regards to security and safety, while others, such as e-health and smart-home [1, 5], are tightly linked to human activities and thus associated with privacy issues.

The security of LoRaWAN has been extensively researched, from denial of service attacks [6] to compromise of cryptographic material [7], but privacy has rarely been studied. Devices in wireless networks use identifiers as a foundation for link-layer protocols. This information can be leveraged to re-identify LoRaWAN devices [8] and infer their activity [9, 10].

LoRaWAN device identifiers are subject to change over time. Based on a year-long real-world trace studied in this paper containing 25k identifiers, around 50% of devices use an identifier lasting less than a week. We also observe that more than a million re-connection processes occur, resulting in identifier rotations.

In addition, identifiers can be randomized, such as Bluetooth Low Energy (BLE) and WiFi MAC addresses [11], reducing the number of messages using the same identifier. Hence, earlier methods such as activity inference [9] will require devices to be observed for each identifier update to be tracked over time. Frequent new identifiers also reduce the efficiency of such attacks [8–10].

We propose an alternative attack methodology of device fingerprinting, which identifies an object by its distinctive characteristics, irrespective of the observation period. The works considering fingerprinting on wireless IoT devices leverage various information, such as extracted traffic characteristics in BLE [12, 13] and WiFi [14, 15]. Contrary to

[1] `https://www.semtech.com/lora`, May 2024.

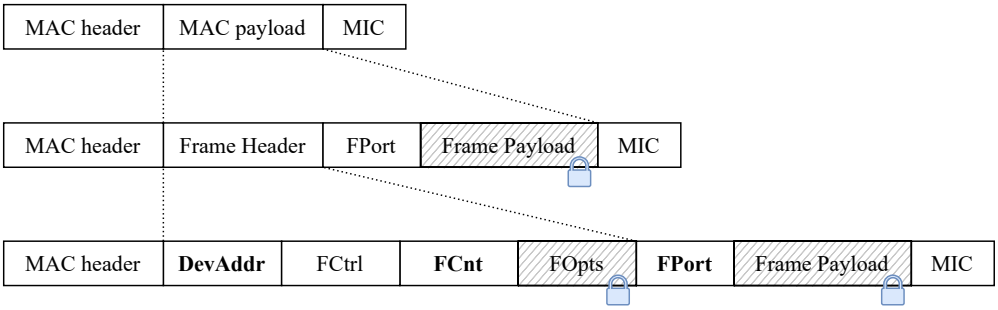


Figure 1: MAC format (relevant unencrypted fields in bold; encrypted fields hatched and marked with a padlock).

existing work in LoRaWAN [8], we study real-world devices while leveraging the full range of features available to an attacker. *To the best of our knowledge, fingerprint-based device tracking has never been studied in LoRaWAN.*

In this work, we investigate the possibility of fingerprinting LoRaWAN end-devices which allows linking or associating various identifiers of the same device. Unlike previous work on wireless networks [8, 12, 13], we compare various fingerprint representations and carefully select a combination of multiple domains: the radio domain, time domain, and content-based domain. Once effective fingerprints are generated, we leverage a machine learning process to predict if two sequences of messages originate from the same end-device. Using large real-world datasets passively collected over the air using off-the-shelf hardware (cf. Section 5), we achieve an accuracy of 0.98, surpassing the performance of the current state-of-the-art methods. Our contributions are fourfold:

1. We present an extensive evaluation of existing fingerprint representations, including vectors, distributions, descriptive statistics, and Markov chains. This allows us to introduce a novel approach called holistic fingerprint utilizing all of the aforementioned methods (cf. Section 6).
2. We leverage such a fingerprint representation to propose a machine learning solution that predicts the origin of sequences of LoRaWAN messages, exploiting multiple domains, including radio, time, and content-based features (cf. Sections 7 and 8).
3. We conduct a large-scale performance evaluation based on real-world datasets composed of up to 41 million messages for multiple scenarios (varying number of received messages or available listening stations, mobile versus static end-devices) (cf. Section 9).
4. Finally, we propose and evaluate a set of potential privacy-preserving countermeasures against the LoRaWAN device linkage. We highlight the difficulty of thwarting fingerprinting without significantly disrupting communications (cf. Section 10).

2. LoRaWAN background

Built upon the LoRa modulation in the physical layer, LoRaWAN is a MAC layer protocol standardized by the LoRa Alliance [16]. It is important to distinguish LoRa, the radio modulation, from LoRaWAN, the higher level MAC layer protocol. In this paper, we focus on the latter but also leverage some radio signal indicators from the lower layer.

2.1. Network architecture

In a typical LoRaWAN architecture, represented in Figure 2, multiple end-devices (e.g., sensors) are connected via radio to one or more gateways. The incoming LoRa communications are translated by gateways into classic IP/TCP or UDP packets and routed to various servers. Once a message is received and processed, the servers have the option to answer the end-devices. In this symmetric communication, a message from end-device to a server is called `uplink`, and `downlink` in the opposite direction. In the following, if unspecified, "message" refers to an `uplink` message.

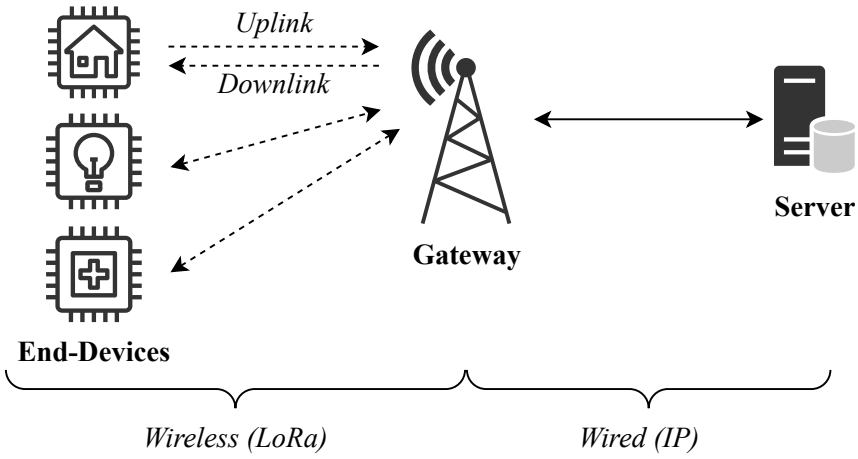


Figure 2: Simplified architecture of a LoRaWAN network.

2.2. Frame structure and identifiers

As seen in Figure 1, a LoRaWAN message is encapsulated within a MAC payload, along with a MAC header, including the message type (see table 2 for examples) and the protocol version. Additionally, the Message Integrity Code (MIC) is used to detect integrity errors during transmission. The MAC payload itself contains the Frame Header, the Frame Port (FPort), and the Frame Payload. Similarly to the TCP/IP model, the port is used to target specific applications: one port can be used for maintenance purposes while another is used for reporting sensor data.

Although the Frame Payload and Options are encrypted, the rest of the header fields are in clear. For example, the `DevAddr` field is a pseudonym assigned to an end-device each time it joins or re-joins the network. The `DevAddr` remains unchanged and can span multiple months, unless a disconnection or a Rejoin process occurs, enabling device tracking across time and space. Likewise, the Frame Counter field (`FCnt`) is incremented for each message, and used by both end-devices and servers to detect packet losses.

3. Related works

3.1. Privacy & security in LoRaWAN

Security of LoRaWAN networks has been considered from its first design and following iterations have corrected identified issues [7, 17, 18]. Threats against LoRaWAN networks include replay attacks [17, 19], denial of service [6], and bit-flipping [20].

So far, privacy issues have seldom been investigated in LoRaWAN. The first privacy threat that has been studied is the inference of activity from LoRa traffic. In [10], variations of signal strength emitted by parking sensors are utilized to detect the presence of a vehicle. Traffic patterns have also been used to infer activity in [9] along with countermeasures such as delay and dummy traffic.

Identifier linkage and re-identification is another privacy aspect of LoRaWAN that has been investigated. In [21], authors demonstrate how the link between `DevAddr` and `DevEUI`, two uncorrelated LoRaWAN identifiers, can be obtained from frame timing. This work is further improved in [8] and [22] using pattern-based approaches.

3.2. Wireless device fingerprinting

Fingerprinting of wireless devices can be studied through various layers. The physical layer is often used to fingerprint wireless devices, by detecting their small hardware imperfection through artifacts in radio signals [23, 24]. Physical layer fingerprinting of LoRa devices has been conducted using convolutional neural networks [25] and

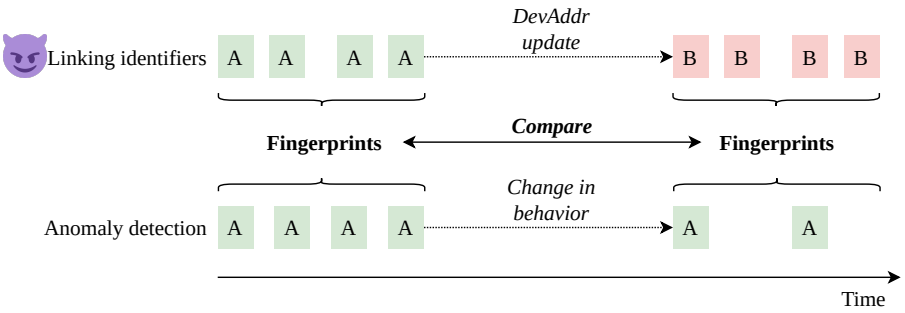


Figure 3: Fingerprinting as a privacy threat and security protection.

other machine learning approaches [24]. However, those approaches require specialized expensive hardware and are evaluated in controlled environments.

Other works opt for upper layers, leveraging various content (length coupled with protocol-specific attributes) or time-based features (inter-arrival time), such as in Bluetooth [26] or WiFi [14]. At a higher level, stateful devices can be represented by Markov chains built from states inferred from network metadata such as timing and size [27, 28]. To the best of our knowledge, the stateful nature of LoRaWAN end-device has never been used for fingerprinting.

Focusing on LoRaWAN, the closest work is the study by Spadaccino et al. [8], that leverages timing patterns to re-identify end-devices. Our solution has significant advantages over their study. First, we do not rely on a synthetic dataset for results and perform our analysis on a large-scale dataset of size 41 million, achieving a higher balanced accuracy of 98%. Second, their solution is analytical and its complexity grows quadratically with the number of active devices. They reduce the time window to just $\sim$ 1 hour following each rejoin, to limit the number of potential devices considered for linking. On the contrary, our method utilizes the whole dataset.

Third, we require only 5 messages per sequence to reach a high accuracy while they leverage all frames from a device. This aids not only an attacker with reduced capabilities but also allows prompt intrusion detection. Finally, they solely rely on time-based features, introducing random delay induces a significant impact on accuracy. We show that a multi-domain approach is more stable against perturbations, as models efficiently adapt to modified/reduced information.

4. Motivations and threat model

Fingerprinting wireless devices can not only compromise privacy but can also enhance network security. Building upon fingerprinting motivations, we proceed to define the threat model illustrated in Figure 3.

4.1. Fingerprinting as a privacy threat

Stable DevAddresses can lead to inventorying devices [29], tracking [30], or to infer personal data [9, 31]. We categorize all these attacks under the general term "linking identifiers", where an attacker associates a sequence of messages with another, for which the corresponding identity or activity is already known. In Figure 3, this corresponds to linking messages from `DevAddr` A with the following ones from `DevAddr` B, as both produce identical fingerprints.

In other protocols, a stable identifier can be replaced by a random and periodically changing one to reduce privacy threats [32, 33]. Although the LoRaWAN specification does not explicitly include address randomization for privacy purposes [34], it does provide a rejoin procedure that can be employed to implement it.[2] Already used in real-world deployments (see Table 2 and [35]), this feature could be adopted widely for privacy as end-devices are increasingly associated with human activities.

[2]The rejoin procedure can be leveraged by an end-device to renew its `DevAddr`. An end-device can use this mechanism anytime on top of its application traffic by sending a "Rejoin-Request message" [16, sec. 6.2.4], at its own initiative. Additionally, a rejoin can be initiated by the server via a "ForceRejoinReq" command [16, sec. 5.13] Such a procedure can also be used to roam between networks or to reset cryptographic keys.

Fingerprinting can thwart address randomization schemes [14]. Hence, defeating randomization schemes through fingerprinting allows attackers to build evermore comprehensive behavioral profiles [9] of devices and users.

4.2. Fingerprinting as a security protection

LoRaWAN is exposed to a wide range of security threats [29]. Such attacks include denial of service or vulnerability exploitation [19, 36, 37], and can have a serious impact on the network infrastructure and associated systems. Furthermore, devices and network keys can be compromised and leveraged to create a fake device [7] launching further attacks.

To counter those threats and build trust in the network, device fingerprinting is a promising approach [19, 38–40]. By building a fingerprint for each legitimate device, one could detect changes in behavior in previously genuine end-devices over long periods and thus identify compromises of elements in the network. As shown in Figure 3, a end-device using `DevAddr` A undergoes a behavioral change potentially due to an attack. This can be effectively detected through comparison with its original fingerprint.

Unlike radio-based solutions that detect jamming and physical-layer attacks [19], our approach enables robust detection at the MAC layer, independent of underlying disruptions. Hence, our solution can be adapted to reinforce [40] the overall security of a LoRaWAN infrastructure.

4.3. Threat model

Combining offensive and defensive fingerprinting (privacy threat and security protection) into a single use case, we outline the capabilities of the fingerprinting actor. To avoid confusion with the defensive use case, we refrain from using the term "attacker". The actor listens to LoRaWAN traffic during an initial period and later attempts to link messages to senders.

For offensive fingerprinting, the actor deploys LoRaWAN listening stations in the area of interest, typically based on cheap and readily available LoRaWAN gateways. As passive eavesdropper, they do not need to authenticate to the network and can freely capture on-air data. Conversely, in defensive fingerprinting, the actor is part of the infrastructure and can directly access this data.

In both cases, data collection is passive, with no injection or modification of wireless messages by the actor. We assume encryption remains unchanged, and the content of encrypted payloads is inaccessible. Although a network administrator could potentially access clear-text payloads for defensive fingerprinting, this scenario is beyond our current scope. Regardless, we obtain a high linkage accuracy in this paper.

5. LoRaWAN datasets

In this section, we present public, open-source LoRaWAN datasets and detail their shortcomings. We then describe our own collection and cleanup process to conduct a large-scale dataset.

5.1. Issues with literature datasets

Current public datasets have two major issues when considering our experiments. First, they may lack specific features, such as port number [41, 42]. Second, they are limited in size and diversity. For instance, `LoED` [43] and `LTAD` [44] datasets respectively offer only around 706,000 and 304,000 relevant `uplink` messages in total. Although we show in Section 9.1 that the attack is still possible on these two datasets, we prefer to collect a much larger `CampusIoT` dataset (see Table 1). We believe that the considerably larger scale of our dataset effectively captures the device heterogeneity of real-world LoRaWAN deployments. It allows for the examination of a wider variety of settings, including a much greater number of gateways and devices.

Dataset	# of uplink messages	# of DevAddresses	# of operators	# of gateways
LoED	706k	5.8k	51	9
LTAD	304k	1.2k	117	3
CampusIoT	**41M**	**24k**	**132**	**49**

Table 1: Datasets summary, with DevAddr and operators receiving more than 5 uplink messages.

Type	Number of messages
Uplink data	113 619 732 (78.1%)
Downlink data	3 384 938 (2.3%)
Join Request	24 402 444 (16.8%)
Join Accept	1 032 577 (0.7%)
Rejoin Request	1 125 902 (0.8%)
Proprietary	1 945 403 (1.3%)

Table 2: Classification of message types.

5.2. CampusIoT dataset

We collect our dataset named CampusIoT using a set of around 50 gateways listening to the EU 863-870MHz ISM band, from January 2022 to September 2022 as part of a research project. These gateways are deployed in the vicinity of Grenoble, France, and operated by the University of Grenoble through the CampusIoT platform. As LoRaWAN messages are broadcast, gateways receive messages from the operator deploying them but also from third parties using the same frequencies. Due to its sensitive nature, the complete dataset is not public.[3]

The dataset comprises 41 million LoRaWAN messages after removing unreliable and irrelevant data from more than 146 million messages (see Section 5.2.1). We focus specifically on third-party operators to study real-world deployments. Thus, it is impossible to precisely know the topology or characterize the deployed end-devices. A summary of the final datasets used for our experiments is available in Table 1. Ours contains more than 130 third-party operators corresponding to 24,000 unique DevAddr. On average, the daily message count is around 140,000.

5.2.1. Selecting relevant data

The deployed gateways do not monitor the downlink messages from third-party operators. We thus focus on uplink messages and clean the dataset from any other type of message showcased in Table 2.

We exclude experimental data associated with known addresses reserved for private and/or experimental end-devices and from known hobbyist operators. Experimental traffic generally exhibits extreme behaviors, unrepresentative of real-world traffic, with end-devices communicating as often as every minute. In the CampusIoT dataset, $\sim$ 10% of uplink messages correspond to experimental addresses.

5.2.2. Studying uplink messages

In this paper, messages are grouped according to their DevAddr. We look at the number of messages sent by a DevAddr in Figure 4.

There is a large fraction (67%) of DevAddr appearing only once due to two possible factors. First, the CampusIoT dataset includes third-party end-devices that may be located at the edge of the gateways' range, with only one of their messages being received. Second, errors or experimental conditions may have caused some addresses to appear only once. In this paper, we disregard DevAddr that appear fewer than 5 times, ultimately retaining 24% of the addresses.

[3] After consulting with our IRB, we received permission to work on third-party traces only if their names were pseudonymized and the dataset kept private. The source code will be open-sourced upon publication.

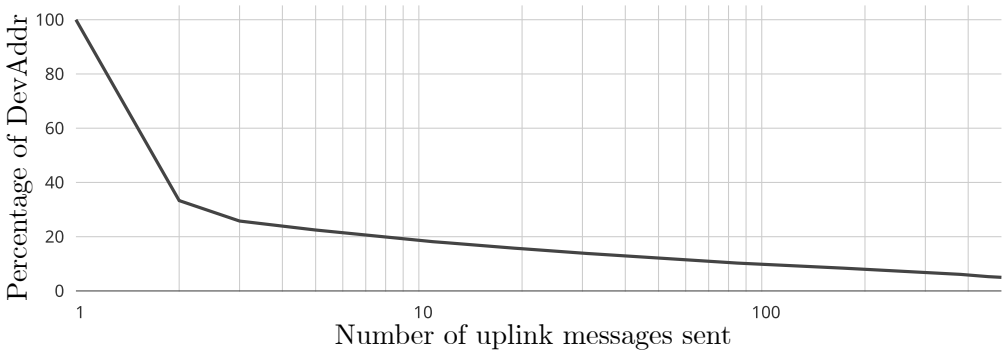


Figure 4: Reverse CDF of `uplink` messages from `DevAddr`.

6. Selecting fingerprinting representations

Picking correct fingerprint representations is an important step in the fingerprinting process. It is important to differentiate between the representation of a fingerprint and its actual content. The *content* refers to features extracted from some dataset, regardless of their domain (e.g., radio or time-based), while the *representation* corresponds to the format used to refine, organize, and exploit these features as a fingerprint. In this section, we present various fingerprint representations used in previous works for network fingerprinting and propose a new holistic approach.

The creation of a fingerprint starts with the extraction of a vector of raw values $v = (x_1, x_2, ..., x_n)$ from network traces. Then, a function $f(.)$ generates a fingerprint $F = f(v)$, which represents a device and is sufficiently distinct to be distinguishable from the fingerprints of other devices.

6.1. Using vectors of values

In its simplest form, the vector of raw values v (extracted from the network traces) itself can be used to fingerprint an object [45–47]. In case of multiple raw features (e.g., length *and* inter-arrival time), the fingerprint is a concatenation of multiple vectors: $F = f(v_1, v_2, ..., v_n) = (v_1, v_2, ..., v_n)$.

6.2. Using distributions

With a significant number of raw features and a high volume of network traffic, vectors may be inappropriate to be used directly by machine learning models. Instead, raw values are aggregated as a distribution, before being transformed into a discrete histogram h_b with n_b bins.

The fingerprint h_b is thus a vector of n_b values representing the heights of the discrete histogram. As it provides a compact representation, and as distribution of values is often enough to characterize a device, it has often been used with time features [26, 48], radio features [23, 49], and content-based features [50].

6.3. Using descriptive statistics

Depending on the studied space, a distribution approach may not be practical. An alternative is to use statistical measures such as mean, variance, standard deviation, skewness, and kurtosis [51]. [4] Here, the fingerprint is equal to a vector of descriptive statistics, such as $F = (\overline{v}, \sigma^2, \sigma, \gamma_1, Kurt)$.

6.4. Using Markov chains

Previous representations do not capture the stateful nature of some network communications. Few fingerprinting works consider raw features as states of a stochastic process [27, 28, 52].

Messages emitted by end-devices may depend on the state of the end-device and its immediate environment. For example, one end-device might send a sensor message once a day at 1 PM, while another may communicate every

[4] `https://en.wikipedia.org/wiki/Kurtosis`

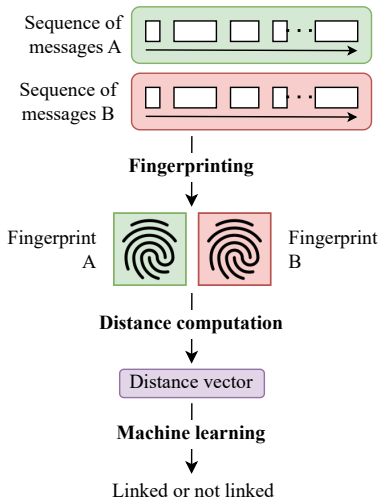


Figure 5: Overview of the linking process.

hour. Thus, a Markov chain can be derived to represent the observed behavior of the end-device. Although this concept has only been leveraged for encrypted web browsing identification in the literature, we argue it can also be applied to identify stateful LoRaWAN end-devices.

We simplify the stochastic process of end-devices communications as a Markov chain that is both of first order (the probability of the next state X_{t+1} only depends on the current state X_t) and homogeneous (the transition probabilities do not vary across time). Thus, the fingerprint corresponds to a stochastic matrix $P = \{p_{i,j}\}$, where $p_{i,j}$ is the probability of transition from state i to state j.

6.5. Our approach: holistic fingerprint

Contrary to previous works, we utilize all fingerprint representations simultaneously, such as:

$$F = (v, h_b, \overline{v}, \sigma^2, \sigma, \gamma_1, Kurt, P)$$

When comparing two fingerprints, the machine learning model automatically selects the most relevant parts of the various representations (see Section 9 for a practical demonstration). Generating such a diverse fingerprint is by design more computationally expensive than single representations. However, the process is done on unconstrained nodes and there is more than enough time to compute the fingerprint, thanks to the low throughput of LoRaWAN networks.

7. Fingerprint-based linkage

In this section, we present a method to link together two sequences of messages generated by the same end-device, as shown in Figure 5. We extract features from various domains to build fingerprints following the representations presented in Section 6. Then, we compare the fingerprints via a distance function.

The resulting distance vector is input into a supervised machine learning model, which classifies it as either linked or not (i.e., originating from the same end-device or from distinct end-devices). We discuss the implementation details and specifics of the machine learning approach in Section 8.

We select a set of raw features extracted from the traffic traces presented in Section 5. Such features belong to three main domains: content-based features[5], time-based features, and radio-based features. Features exhibiting diversity

[5]The term content corresponds to the bytes transmitted over the air and not the actual data carried by the message (which is unavailable because of encryption).

Name (unit)	Range	Domain
Port number	[0; 255]	Content
Payload length (bytes)	[0; 222]	Content
Arrival time (s)	-	Time
Inter-arrival Time (s)	-	Time
RSSI (dBm)	[0; −139]	Radio
SNR	[−26; 30.5]	Radio
ESP (dBm)	[0; −160]	Radio

Table 3: List of raw features.

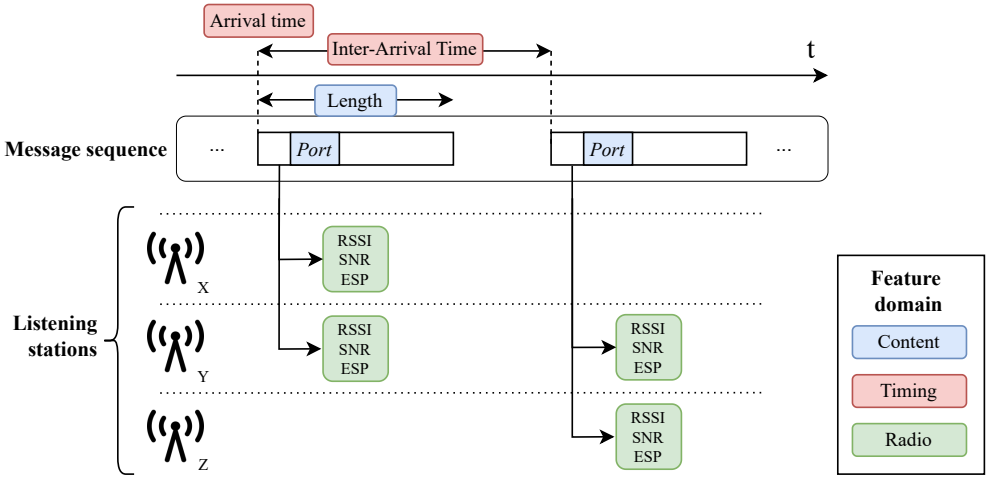


Figure 6: Extracting raw features from a sequence of LoRaWAN messages.

across different end-devices are selected to generate fingerprints unique to each end-device. Intuitively, combining all domains should enable us to identify individual devices (showcased in Figures 8b and 10).

The raw features considered in this work are presented in Figure 6, listed in Table 3, and discussed in detail in the following sections.

7.1. Content-based features

Content-based features are extracted from the clear-text sections of messages. As discussed in Section 2, specific fields in the message headers are not encrypted and provide valuable information about the state of the communication. In addition to the `DevAddr`, these fields include the *Port number* and *Payload length*, which are further described below.

Port number. Represented as an integer between 0 and 255, the port number is generally used to target a specific application. Its value can depend on the nature of the message (e.g., a port can be selected for maintenance purposes, while another one is assigned to sensor data transfer). More precisely, 0 is reserved for MAC commands, values from 1 to 223 are application-specific, and port numbers over 224 are reserved for the LoRaWAN MAC layer test protocol [16, sec 4.3.2]. To construct a fingerprint, the value of this field, as well as its evolution within a sequence of messages, is relevant: the maintenance port may be used solely after the join process or throughout the communication, depending on the type of end-device.

Payload length. Represented as an integer between 0 and 242, the *payload length* is extracted as metadata indicating the number of bytes composing the payload. The payload is optional and the upper bound may change in other regions [53]. Since LoRaWAN messages are not padded, the length of the encrypted payload provides an indication of the size of the transmitted data. The length of messages can vary w.r.t. the application, as well as across a message

sequence. For instance, a sensor sends hourly readings of a constant size but may occasionally transmit maintenance reports of a different length.

7.2. Time-based features

Time-based features are metadata corresponding to temporal information associated with message transmission. Two time-based features are considered: arrival time and inter-arrival time.

Arrival time. Represented as an integer Unix timestamp with an accuracy of a millisecond, arrival time corresponds to the time of reception of a message by a listening station. From this, additional time-based features can be derived, such as the hour of the day (e.g., 3 PM) or the day of the week (e.g., Monday) when a message was received.

Inter-Arrival Time (IAT). The time difference between the reception of two consecutive messages is called Inter-Arrival Time. As opposed to previous features, the IAT is associated with two messages.

Both arrival time and IAT can vary from one end-device to another, as they depend on the nature of the application and on the internal state of the end-device. For instance, an end-device may transmit messages every hour and could be configured to transmit a maintenance message every Monday at 12 PM.

7.3. Radio-based features

The final set of raw features contains indicators providing information about the radio layer. In case the fingerprint actor uses more than one listening station, radio-based features may differ from one to another and thus form a multidimensional vector; such a format is applied to the following features: **Received Signal Strength Indication** (RSSI), **Signal-to-Noise Ratio** (SNR) and **Estimated Signal Power** (ESP).

While RSSI represents the received signal power, SNR is the ratio between the received power signal and the noise floor power level. Derived from both RSSI and SNR, ESP is used to compare the channel quality in radio communications. ESP provides more precise values than the RSSI when its value drops low (∼-120dBm) [54].

These radio-based features remain relatively stable when considering the case of geographically static devices [55]. Except for scenarios involving tracking sensors in mobile objects, the environment surrounding an end-device is not subject to major changes. Thus, two sequences of messages originating from the same end-device should exhibit similar radio-based features. Since radio-based features are not linked to the internal state of the end-device, they are the only features in the fingerprint that lack a Markov chain representation.

8. Evaluation methodology

In this section, we present the dataset generation along with its *ground-truth*. We also detail the machine learning specifics and the performance metrics utilized. We follow the same process for both the `CampusIoT` and the literature datasets.

8.1. Dataset generation

We extract sequences of messages from the LoRaWAN traffic, labeled by the corresponding `DevAddr`. Since two distinct end-devices may use the same re-assigned `DevAddr` at different times, we enforce a maximum period of inactivity T, after which the `DevAddr` is considered to belong to a new end-device. We set $T = 1$ week, assuming that the majority of end-devices send at least one message per week.

For each set of messages, the `FCnt` values are verified, and any sequence of messages exhibiting an unexpected order is discarded. Each sequence is then processed to produce a fingerprint formatted as specified in Section 6.5 using features presented in Section 7. To compare two fingerprints F_1 and F_2, a distance vector is computed, matching each feature of F_1 against its equivalent in F_2. We use the Euclidean distance for statistical aggregates features (e.g., average or distributions), and the Euclidean norm (or l^2-norm) for matrices representing Markov chains.

The dataset employed to train the machine learning model is created from two sets of fingerprint pairs: linked pairs correspond to sequences emitted by the same end-device, and not linked pairs correspond to sequences emitted by distinct end-devices. Following a one-vs-all approach, we generate fingerprint pairs by randomly sampling other fingerprints. For each fingerprint, we create the maximum possible number of linked pairs, along with up to 5 not linked pairs. By design, this results in an imbalanced dataset (see Section 8.4). We consider this setting more realistic,

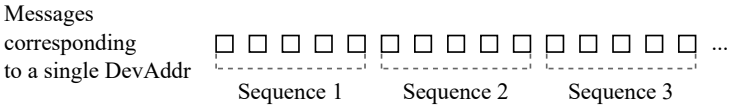


Figure 7: Splitting messages coming from the same device.

as network traces are inherently imbalanced, containing more traffic from different end-devices than from the one currently being fingerprinted. Each pair is accurately labeled based on the relationship between the input sequences (linked or not linked).

8.2. Ground-truth: labeling sequences

LoRaWAN traffic generated by third-party operators (`CampusIoT` dataset) or provided in literature datasets (`LTAD` [44] and `LoED` [43]) does not explicitly indicate whether two fingerprints originate from the same end-device. To construct a ground truth of linked and not linked fingerprint pairs, we propose segmenting sequences based on an already known identifier, namely the `DevAddr`, as illustrated in Figure 7.

For each set of messages S associated to an end-device via its `DevAddr`, the set is divided into chronologically ordered sequences $S_1, S_2, ..., S_m$ each containing c messages. Thus, a `DevAddr` corresponding to M messages corresponds to $\lfloor \frac{M}{c} \rfloor$ sequences. In practice, we select cardinalities of 5, 10, 20, 50, 100, and 200 for c; so if 233 messages are received, a total of 46, 23, 11, 4, 2, and 1 sequences of lengths 5, 10, 20, 50, 100, and 200 are respectively created.

Thanks to the `DevAddr` identifying the origin of each sequence, we are able to accurately label distance vectors presented in Section 8.1, providing us with a ground truth. In practice, end-devices that generate a large number of messages may also produce a high number of sequences. As a result, these end-devices can become over-represented, potentially skewing the machine learning dataset. This is especially relevant when considering sequences of short length. To avoid this, the maximum number of sequences is limited to 3 per end-device, so $|S| \leq 3$.

8.3. Linkage

Given two distinct sequences of messages, the goal of the linkage is to predict whether they originated from the same end-device. The linkage process applies the previously described method to a pair of sequences: fingerprints are generated for each sequence, the corresponding distances computed, resulting in a vector fed to the pre-trained machine learning model. The model finally produces a binary prediction: linked or not linked.

8.4. Machine learning specifics

Following the state-of-the-art [22, 31], we initially tested several well-studied classification algorithms, including AdaBoost, decision tree, LightGBM, logistic regression, naive Bayes, k-nearest neighbours, and random forest. As shown in other works [22], the random forest algorithm outperformed the others during our preliminary experiments; therefore, we selected it for further analysis throughout the remainder of the paper. Nonetheless, our goal is to demonstrate that fingerprinting is feasible. Further optimizations or alternative classification algorithms may yield improved results; we leave such investigations to future work.

We conduct a 5-fold cross validation, splitting the dataset presented in Section 5 into 80% for training and 20% for testing [56]. In order to reduce the risk of test snooping [57] (leveraging the testing dataset for something other than evaluating the model), we create two subsets based on the initial `DevAddr` value. Thus, fingerprints coming from the same end-devices (i.e. same `DevAddr`) are only present either in the training *or* the testing dataset.[6]

The dataset contains a significantly higher proportion of incorrect pairs compared to correct ones (75% incorrect vs. 25% correct), which may lead to overfitting. To address this imbalance in the training dataset [58], the minority class is over-sampled using SMOTE [59, 60].

[6] Although this process correctly separates fingerprints coming from the same `DevAddr`, it does not protect from end-devices changing their `DevAddr` and randomly having fingerprints from each in the training *and* the testing dataset. As re-join processes are rare (see table 2) and as it is not possible to detect such a behavior due to using third-party operators traces, we consider that such an unlikely event does not hinder the overall process.

Dataset	CampusIoT	LoED	LTAD
Balanced accuracy	0.9436	0.9330	0.9485

Table 4: Performance w.r.t. datasets.

The *testing* dataset remains unchanged to prevent any data leakage [61], as SMOTE's resampling process may inadvertently leverage specific samples that could be used for testing. Maintaining the testing dataset in its original form ensures that it reflects the natural imbalance of the observed data and helps in identifying potential model biases.

8.5. Performance metrics

We use the balanced accuracy (*BA*), to tackle the effects of a class imbalances. *BA* represents the mean of the sensitivity and specificity, correctly taking into account the class imbalance of the dataset. The higher the balanced accuracy is, the better the model is at predicting both classes, with 0.5 corresponding to a random prediction, and 1 representing perfect predictions. When applicable, axes are limited from 0.7 to 1 balanced accuracy for a better clarity.

Balanced accuracy gives an overall assessment of model performance and hence is used by us across the paper. Though, it may hide asymmetries in classification errors. Given the large number of samples, a model could achieve a high balanced accuracy while still misclassifying a significant number of samples. To alleviate these concerns, we complement the balanced accuracy by providing the True Positive Rate (TPR) and the True Negative Rate (TNR), either in the main body of the paper, or in Appendix A for completeness.

Finally, we repeat the whole evaluation process for 15 seeds when initializing the PRNGs and provide a 95% confidence interval.

9. Results

This section evaluates the proposed scheme using real-world datasets with up to 41 million messages (see Section 5), utilizing all available listening stations unless stated otherwise. We start by evaluating the effectiveness of fingerprint on each dataset Then, we study the impact of fingerprint representations and various feature domains, as well as the number of listening stations.

9.1. Fingerprinting devices in multiple datasets

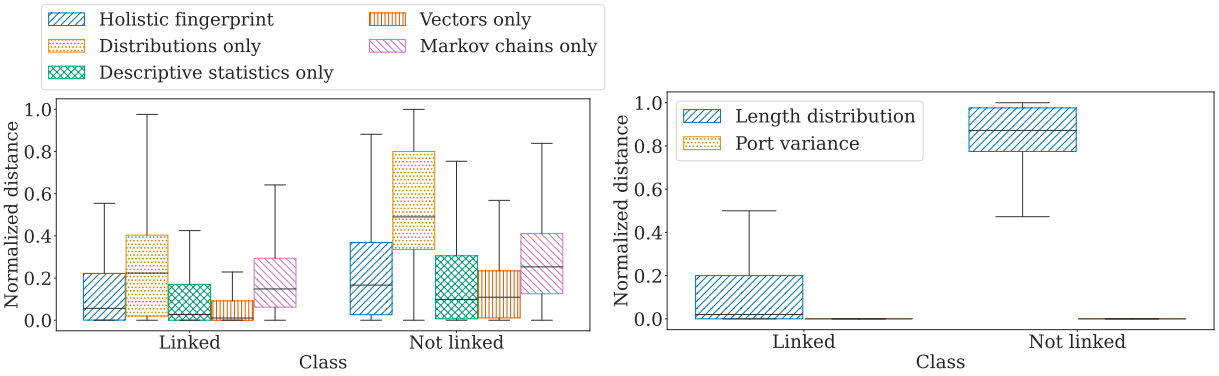
Open-source datasets provide fewer `uplink` messages compared to `CampusIoT`, resulting in fewer sequences of 20 messages or more. For instance, we can only generate 138 and 233 fingerprints for 200 messages per sequence with the `LTAD` [44] and `LoED` [43] datasets respectively. In contrast, the `CampusIoT` dataset allows us to generate 9,855 fingerprints for the same sequence length. This disparity in dataset size limits the ability of machine learning models trained on the `LTAD` [44] and `LoED` [43] datasets to generalize effectively and yield significant results.

We select sequences of five messages to generate a sufficiently large set of fingerprints, ensuring robust training for the machine learning models. We observe in Table 4 that all three datasets reach more than 0.93 balanced accuracy, showing that fingerprinting LoRaWAN devices is indeed possible in a variety of geographical settings and devices.

For the remainder of the paper, we will use only the `CampusIoT` dataset. This allows us to comprehensively evaluate the attack in diverse settings with a greater number and range of devices.

9.2. Evaluating fingerprint representations

The quality of a fingerprint depends on two characteristics. A good fingerprinting process produces similar values from information originating from the same device (consistency), and apply significantly different values when applied to two distinct entities (discriminatory power) [62]. Said otherwise, linked pairs of fingerprints should have a high consistency, producing minimal distances. On the other hand, not linked pairs should produce distinctively greater distances. We explore this concept in Figure 8 for the `CampusIoT` dataset.



(a) Distances between fingerprint pairs w.r.t. fingerprint representation.

(b) Distances between fingerprint pairs for two extreme feature examples.

Figure 8: Evaluating fingerprint representations.

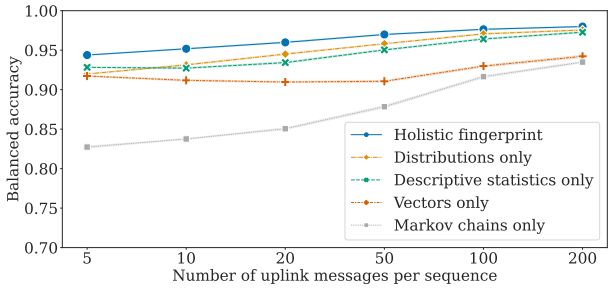


Figure 9: Performance w.r.t. fingerprint representations.

Figure 8a shows the distribution of distances for each fingerprint representation, for all lengths of sequences. As two fingerprints produce a vector of distances of various orders of magnitude, we first normalize the values. At first glance, there is a notable difference of distributions between linked and not linked pairs: linked pairs generally yield lower distances. Distribution-based fingerprints show a significant heterogeneity of distances, with not linked pairs clearly above their counterparts.

Similar tendencies are seen on a per-feature basis. As showcased in Figure 8b, some features exhibit much better consistency and discriminatory power than others. For example, the length distribution is different in linked and not linked pairs, whereas the port variance is almost the same.

Although this suggests a potentially exploitable difference between fingerprint pairs, it does not reveal the actual performance variations during classification. Thus, we train models using only one fingerprint representation and compare their balanced accuracy.

As seen in Figure 9, the *holistic fingerprint* provides better results than other representations from the literature. Notably, *distributions* and *descriptive statistics* yield comparable results for sequences of more than 100 `uplink` messages, suggesting that with enough data, these simpler representations remain relevant.

Radio-based features are considered stateless and thus not represented as Markov chain. Additionally, the corresponding statistical state transitions matrix requires enough messages to stabilize. This can explain why the *Markov chains* representation initially produces a comparatively low balanced accuracy compared to other representations. However, we find that removing it from the *holistic fingerprint* still reduces the balanced accuracy slightly so we keep it. Other results presented in this paper are generated using the *holistic fingerprint* representation.

9.3. Sequence length and feature domain's impact

The first factor to consider is the length of the sequences to be linked. Specifically, we evaluate the impact of the number of messages and the feature domains composing these sequences. As detailed in Section 8.2, we consider sequence lengths of 5, 10, 20, 50, 100, and 200 messages.

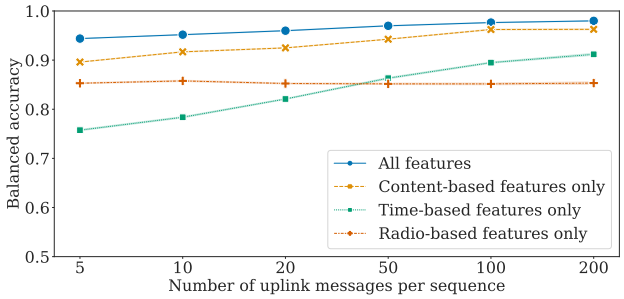
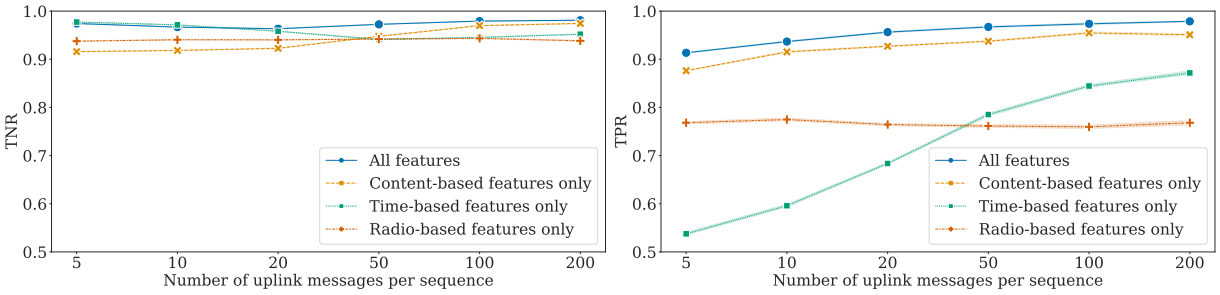


Figure 10: Performance w.r.t. feature domains.



(a) TNR.

(b) TPR (y-axis length expanded for clarity).

Figure 11: Performance w.r.t. feature domains.

Figure 10 represents the balanced accuracy based on different lengths, leveraging *all features* as well as fingerprints using only one of the three features domains. For the *all features* setting, the performance is high across all sizes of sequences, with a balanced accuracy starting at 0.94 for 5 messages up to 0.98 balanced accuracy for 200 messages per sequence. The performance improvements as sequence length grows is expected: fingerprints become more accurate as the number of samples grows.

Figure 10 also illustrates the performance when only a single domain of features is utilized. Overall, content-based features alone deliver the best performance, starting at 0.90 for a sequence length of 5 and improving to achieve a level comparable to the *all features* configuration for sequence lengths greater than 100. Performances obtained with time-based features only start at 0.76 and gradually increase with the length of the sequence to a level below that of the *all features* setting.

Time-based features start lower than other domains at 0.78 balanced accuracy for 5 messages per sequences, yet reach 0.91 for 200 messages per sequences. We hypothesize that the drop in accuracy is due to: a) similar temporal patterns utilized by multiple devices, b) high sensitivity of timing information from noise related to the data capture, as well as the device itself. While the first factor reduces the discriminatory power of time-based features, the second one leads to higher instability.

When using only radio-based features, the balanced accuracy starts at 0.85 and remains stable as the sequence length increases. This could be explained by a side effect of the selection of end-devices having sufficiently long sequences of messages (their radio features may be less reliable than those of other end-devices).

In some applications such as network security monitoring [40], the linkage should be done as soon as possible. Early detection enables timely intervention, before devices can further disrupt network performance or compromise security. Due to the low transmission rate of LoRaWAN networks, the median time to receive 5 messages from the same device in the `CampusIoT` dataset is approximately 6 hours. Despite the extended time frame, the ability to detect misbehaving or compromised end-devices after just 5 messages is potent.

To further analyze the models adaptation to sequence lengths, we study the TPR and TNR in Figures 11a and 11b. The relatively stable TNR suggests that negative samples are consistently identified, while the increasing TPR indicates improved detection of positive cases with longer sequences. These observations support the robustness of our

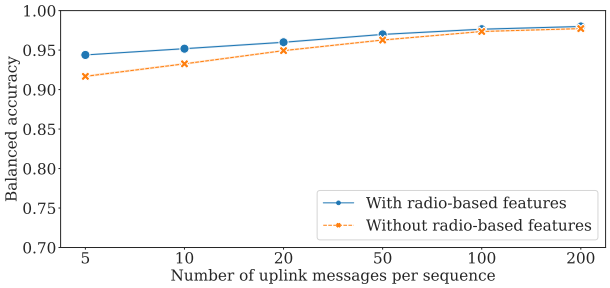


Figure 12: Performance w.r.t. radio-based features.

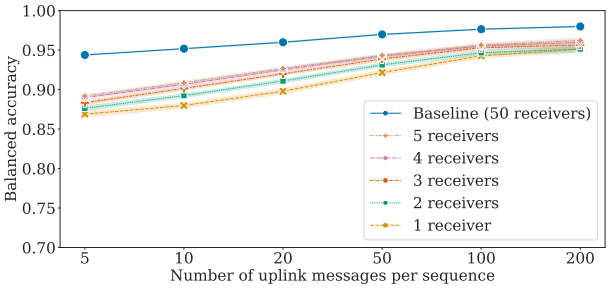


Figure 13: Performance w.r.t. controlled listening stations.

approach across different sequence lengths when utilizing all or content-based features.

However, they also highlight a key limitation: short sequences are subject to numerous false negatives when relying on other feature domains. In particular, *time-based features only* fail to reliably link short communications, starting at 0.54 TPR with a sequence length of 5 and increasing up to 0.87 with a length of 200. This implies that an attacker would need to eavesdrop over extended periods to perform effective fingerprinting using time data alone. As discussed in Section 10, this limitation presents an opportunity to effectively counter fingerprinting attacks.

Summary: When utilizing all features, performance is high and improves as sequence length increases. More surprisingly, content-based features alone are sufficient for reliable linkage as shown in Figures 10 and 11. Said otherwise: time and radio-based features are *not* required to reach high fingerprinting accuracy. Finally, although the TNR is stable across sequence lengths, the TPR increases with the number of available messages, notably for time-based features, indicating an interesting path for countermeasures.

9.4. Radio-less fingerprinting

Radio-based fingerprinting is less useful for mobile end-devices as context variations (distance, environment, obstacles) may modify the channel significantly. It is impossible to determine the mobility of an end-device based only on radio-features fluctuation. Hence, we consider the extreme case where radio-based features are not available.

Figure 12 shows that, for sequences of 5 messages, the balanced accuracy decreases from around 0.94 to 0.92 (-2.87%), and for sequences of 200 messages, it reaches nearly the same value (around -0.30%). Thus, even if the performance is slightly lower, our fingerprinting scheme could be applied to a scenario where radio features are unavailable or unreliable, which is the case for mobile end-devices.

9.5. Impact of controlled listening stations

Radio-based features enhance classification accuracy, particularly for short sequences. So far, the results are based on all messages, regardless of the number of capturing listening stations. Each listening station controlled by the attacker provides a portion of radio-based information; increasing their number also increases the global amount of information from this domain.

We study nested datasets corresponding to different numbers of listening stations: the n-listening-station dataset includes all the messages of the $n-1$-listening-station dataset. We choose $n = 5$ due to the computational complexity and utilize the highest receiving listening stations.

As seen in Figure 13, for sequences of 5 messages, performance improves by 9.43% when increasing the number of listening stations from 1 to 5. In contrast, for sequences of 200 messages, the improvement is only 7.96%. The number of listening stations has a significant impact on performance, with balanced accuracy increasing as more stations are utilized. However, this impact diminishes with longer sequences, as other domains such as time and content provide more information. In a real-world scenario, fingerprinting remains effective even with a small number of strategically deployed stations.

10. Countermeasures

In this section, we consider ways to counter the linkage approach. We study the possibility of reducing the performance of the linking scheme by altering the behavior of end-devices to disrupt the features exploited by the machine learning process. We start with introducing potential countermeasures before evaluating their effectiveness.

Countermeasures are only relevant to the offensive fingerprint use case, where the behavior of end-devices can be modified to thwart an external fingerprinting attack. Modifying the end-devices to reduce *fingerprintability* in the defensive use case would degrade the network's protection.

Additionally, we note that dummy traffic [63, 64] generate a high overhead and is not suited for the highly energy-constrained LoRaWAN protocol [65]. Thus, we do not explore this path.

10.1. Suggested countermeasures

1. **Obfuscating message length using padding:** One of the main features used by our approach is the length of the message, and an obvious way to obfuscate it is to apply padding. The amount of padding that can be added to a message is limited by the maximum size of the payload (between 51 and 222 bytes), which depends on the data rate (DR). We find that 99.82% of messages in the `CampusIoT` dataset have a payload shorter than 51 bytes, which leaves room for padding even in the most conservative cases. Padding can be done either by filling up the payload to its maximum size or for a uniformly selected random length ($\leq$ the available space). Uniform padding guarantees Approximate Differential Privacy, a relaxed version of Differential privacy [66].
2. **Obfuscating the port number:** The port number is an indicator exposed in clear and exploited by our fingerprinting approach. One way to obfuscate it is to encrypt it as the rest of the payload. We can use cryptographic primitives already available in LoRaWAN. More specifically, we propose to encrypt the port number the same way the header options are. As the port is not obligatorily used before the decryption, the lack of availability of this value will have no impact on the message processing. [7]
3. **Introducing random delays:** Time-based data plays a significant role in fingerprinting end-devices, as demonstrated in Figure 10. Thus, disrupting temporal patterns is a natural approach to counter fingerprinting. One effective method for altering these temporal features is by introducing random delays before transmitting messages [9], such as at the application layer.
4. **Modulating the transmit power:** Radio-based features are another source of information used in our attack. Disrupting those features is challenging and can be achieved by dynamically modifying the transmit power of the end-device. Indeed, radio-based indicators (RSSI, SNR, and ESP) all depend on the transmit power. In LoRa modules, transmit power can be adjusted from -4dBm to 20dBm [67]. The steps of those adjustments can be as small as 1dBm and as high as 20dBm. Therefore, adjusting the transmit power using the full range offered by the hardware, would offer an amplitude of 24dBm on the radio features.
5. **Increasing `DevAddr` rotation frequency:** The length of the sequences used to build the fingerprint has a significant impact, as shorter sequences yield lower performances (see Section 9.3). One way to prevent the attacker from using long sequences is to enforce the renewal of the `DevAddr`. Even if identifier rotation is not currently in use, it can be implemented per the current specifications (see Section 4.1) [34]. We virtually simulate `DevAddr` rotations by restricting sequences to various lengths.

10.2. Evaluation methodology

For more impact, we evaluate the combined effectiveness of all countermeasures based on the decrease in balanced accuracy. The baseline is obtained without any countermeasure applied. We consider two perturbation settings with different levels of fluctuations. In both, the port is encrypted and we select uniform padding to avoid extreme overheads.

The *moderate* perturbation involves countermeasures that moderately affect communication, including 10 bytes of uniform padding amplitude, a transmit power variation of [0; 10]dBm, and a random delay of up to 10 minutes. In

[7] For example, the Chirpstack open-source implementation of a Network Server supports decrypting the payload before leveraging the port `https://github.com/chirpstack/chirpstack/blob/318f0973440cc4d80695e7d2e7ac265dea313782/lrwn/src/phy_payload.rs#L858`

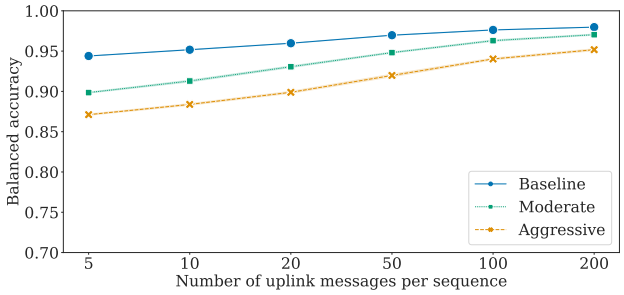


Figure 14: Impact of countermeasures on performance.

aggressive perturbation, countermeasures can significantly degrade communication performance, including 40 bytes of uniform padding amplitude, a transmit power shift of $[-4; 20]$dBm, and a random delay of up to 1 hour. Although 1 hour is theoretically not the maximum amount of delay, we refrain from using greater values to remain within the realm of the possible.

10.3. Countermeasures' effectiveness

As seen in Figure 14, we find that the balanced accuracy for 50 messages per sequence decreases by -2.24% for moderate settings, and by -5.10% for aggressive settings. Using 5 messages per sequence yields a -7.32% in balanced accuracy for moderate countermeasures, and -10.22% in an aggressive setting.

Additionally, the impact of countermeasures is more pronounced in scenarios with a limited number of listening stations (see Section 9.5). For example, adding radio modulation to sequences of 50 messages reduces performance by 0.03% for all listening stations, but by 2.50% when only 3 stations are available in the moderate setting. This suggests that countermeasures may have a greater effect against less powerful attackers.

10.4. Overhead and impact on communications

We note that none of the proposed countermeasures actually breaks the LoRaWAN standard and rather enhances its current design. However, implementing these countermeasures does have inherent impacts.

Expanding the payload size inevitably implies an overhead, as lengthier messages consume more energy and occupy the shared medium longer. In LoRaWAN, this overhead is measured by the Time On Air (TOA), i.e. time required for an end-device to transmit data [68], which is directly linked to battery depletion due to the high energy cost of communications [69]. Unsurprisingly, padding to the maximum payload length increases the TOA by approximately 96%, while uniform padding introduces overheads ranging from about 5% to 46% for padding amplitudes between 5 and 200 bytes, respectively.

Comparatively, encrypting the port number represents a low-cost operation. First, it does not increase the TOA by preserving the length of the field thanks to AES-128 CCM*, already present in the LoRaWAN standard, notably to encrypt payloads. Second, the port number can be added to other to-be-encrypted fields, such as MAC commands. We have found that it would generate an additional low-cost encryption operation only in 4% of the 14 million `uplink` messages containing commands.

However, not all countermeasures induce an easily measurable impact. For instance, while delay potentially disrupts traffic, it is hard to measure its exact impact without knowing the requirements of underlying applications [70, 71]. Additionally, estimating the impact of power modulation is challenging. First, using negative power modulation raises the risk of message loss, and unknown communication conditions prevent network traces alone from accurately predicting message loss. Second, higher transmit power increases energy consumption, depleting end-devices batteries. Although some studies address LoRaWAN radio parameters and energy use [67], further research and real-world deployments are needed to assess potential losses and increased energy consumption.

Summary: Despite significant overheads, such as a 40.69% increase in TOA, the -10.22% drop in balanced accuracy is notable. Given the potential for a greater impact in real-world scenarios (e.g., with fewer listening stations), these countermeasures should be implemented wherever feasible. Starting with simpler measures, such as port encryption, is advisable while still adhering to deployment requirements.

11. Conclusion

We present a study on fingerprinting LoRaWAN end-devices using off-the-shelf hardware, leveraging features from multiple network traffic domains: radio signal, timing, and message content. By proposing a novel holistic fingerprint representation, we reliably predict whether two message sequences originate from the same end-device. Our experimental evaluation on real-world datasets, comprising up to 41 million messages, demonstrates high performance (balanced accuracy of 0.98), even with a single listening station or when tracking mobile end-devices with unstable radio-based features. We also evaluate countermeasures, which marginally decrease fingerprinting performance by -10.22% balanced accuracy but introduce significant overheads.

While we advocate for further privacy enhancements in LoRaWAN, particularly identifier randomization, this work lays the foundation for effective intrusion detection systems, supporting long-term monitoring of LoRaWAN and other LPWAN networks.

Declaration of competing interest

The authors declare no conflict of interest.

Data availability

The data that has been used is confidential.

Acknowledgements

This work has been supported by the ANR-BMBF PIVOT project (ANR-20-CYAL-0002), H2020 SPARTA project and the INSA-Lyon SPIE ICS IoT Chair.

Additionally, the computations were performed using the Grid5000 platform [72].

References

[1] M. Jouhari, N. Saeed, M.-S. Alouini, E. M. Amhoud, A survey on scalable LoRaWAN for massive IoT: Recent advances, potentials, and challenges, IEEE Communications Surveys & Tutorials 25 (3) (2023) 1841–1876.

[2] M. Abbasi, S. Khorasanian, M. H. Yaghmaee, Low-Power Wide Area Network (LPWAN) for Smart grid: An in-depth study on LoRaWAN, in: 2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI), IEEE, Tehran, Iran, 2019, pp. 022–029. doi:10.1109/KBEI.2019.8735089.

[3] M. Sidorov, P. V. Nhut, Y. Matsumoto, R. Ohmura, LoRa-Based Precision Wireless Structural Health Monitoring System for Bolted Joints in a Smart City Environment, IEEE Access 7 (2019) 179235–179251. doi:10.1109/ACCESS.2019.2958835.

[4] O. T. Sanchez, J. M. Fernandes, A. Rodrigues, J. S. Silva, F. Boavida, J. E. Rivadeneira, A. V. de Lemos, D. Raposo, Green Bear - A LoRaWAN-based Human-in-the-Loop case-study for sustainable cities, Pervasive and Mobile Computing 87 (2022) 101701. doi:10.1016/j.pmcj.2022.101701.

[5] A. Mdhaffar, T. Chaari, K. Larbi, M. Jmaiel, B. Freisleben, IoT-based health monitoring via LoRaWAN, in: IEEE EUROCON 2017 -17th International Conference on Smart Technologies, 2017, pp. 519–524. doi:10.1109/EUROCON.2017.8011165.

[6] E. van Es, H. Vranken, A. Hommersom, Denial-of-Service Attacks on LoRaWAN, in: Proceedings of the 13th International Conference on Availability, Reliability and Security, ARES 2018, Association for Computing Machinery, New York, NY, USA, 2018, pp. 1–6. doi:10.1145/3230833.3232804.
URL https://dl.acm.org/doi/10.1145/3230833.3232804

[7] E. Aras, G. S. Ramachandran, P. Lawrence, D. Hughes, Exploring the Security Vulnerabilities of LoRa, in: 2017 3rd IEEE International Conference on Cybernetics (CYBCONF), 2017, pp. 1–6. doi:10.1109/CYBConf.2017.7985777.

[8] P. Spadaccino, D. Garlisi, F. Cuomo, G. Pillon, P. Pisani, Discovery privacy threats via device de-anonymization in LoRaWAN, in: 2021 19th Mediterranean Communication and Computer Networking Conference (MedComNet), 2021, pp. 1–8. doi:10.1109/MedComNet52149.2021.9501247.

[9] P. Leu, I. Puddu, A. Ranganathan, S. Čapkun, I Send, Therefore I Leak: Information Leakage in Low-Power Wide Area Networks, in: Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks - WiSec '18, ACM Press, Stockholm, Sweden, 2018, pp. 23–33. doi:10.1145/3212480.3212508.
URL http://dl.acm.org/citation.cfm?doid=3212480.3212508

[10] L. Dujić Rodić, T. Perković, M. Škiljo, P. Šolić, Privacy leakage of LoRaWAN smart parking occupancy sensors, Future Generation Computer Systems 138 (2023) 142–159. doi:10.1016/j.future.2022.08.007.
URL https://www.sciencedirect.com/science/article/pii/S0167739X22002680

[11] G. Celosia, M. Cunche, Saving private addresses: An analysis of privacy issues in the bluetooth-low-energy advertising mechanism, in: Proceedings of the 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, ACM, Houston Texas USA, 2019, pp. 444–453. doi:10.1145/3360774.3360777.

[12] T. Gu, P. Mohapatra, BF-IoT: Securing the IoT Networks via Fingerprinting-Based Device Authentication, in: 2018 IEEE 15th International Conference on Mobile Ad Hoc and Sensor Systems (MASS), 2018, pp. 254–262, iSSN: 2155-6814. doi:10.1109/MASS.2018.00047.

[13] G. Gagnon, S. Gambs, M. Cunche, RSSI-based Fingerprinting of Bluetooth Low Energy Devices, in: SECRYPT 2023 - 20th International Conference on Security and Cryptography, 2023, p. 1.

[14] M. Vanhoef, C. Matte, M. Cunche, L. S. Cardoso, F. Piessens, Why MAC Address Randomization is not Enough: An Analysis of Wi-Fi Network Discovery Mechanisms, in: Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, ASIA CCS '16, Association for Computing Machinery, New York, NY, USA, 2016, pp. 413–424. doi:10.1145/2897845.2897883.

[15] A. K. Mishra, A. C. Viana, N. Achir, Bleach: From wifi probe-request signatures to mac association, Ad Hoc Networks (2024) 103623.

[16] L. A. T. Committee, LoRaWAN® Specification v1.1, https://lora-alliance.org/resource_hub/lorawan-specification-v1-1/ (2017).

[17] S. Tomasin, S. Zulian, L. Vangelista, Security Analysis of LoRaWAN Join Procedure for Internet of Things Networks, in: 2017 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), 2017, pp. 1–6. doi:10.1109/WCNCW.2017.7919091.

[18] I. Butun, N. Pereira, M. Gidlund, Security risk analysis of lorawan and future directions, Future Internet 11 (1) (2018) 3.

[19] S. Saxena, A. Pandey, S. Kumar, RSS based multistage statistical method for attack detection and localization in IoT networks, Pervasive and Mobile Computing 85 (2022) 101648. doi:10.1016/j.pmcj.2022.101648.

[20] J. Lee, D. Hwang, J. Park, K.-H. Kim, Risk analysis and countermeasure for bit-flipping attack in LoRaWAN, in: 2017 International Conference on Information Networking (ICOIN), 2017, pp. 549–551. doi:10.1109/ICOIN.2017.7899554.

[21] L. Ancian, M. Cunche, Re-identifying addresses in LoRaWAN networks (2020) 28.

[22] S. Pélissier, M. Cunche, V. Roca, D. Donsez, Device re-identification in LoRaWAN through messages linkage, in: Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks, 2022, pp. 98–103.

[23] T. D. Vo-Huu, T. D. Vo-Huu, G. Noubir, Fingerprinting Wi-Fi Devices Using Software Defined Radios, in: Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks, ACM, Darmstadt Germany, 2016, pp. 3–14. doi:10.1145/2939918.2939936.

[24] P. Robyns, E. Marin, W. Lamotte, P. Quax, D. Singelée, B. Preneel, Physical-layer fingerprinting of LoRa devices using supervised and zero-shot learning, in: Proceedings of the 10th ACM Conference on Security and Privacy in Wireless and Mobile Networks, ACM, Boston Massachusetts, 2017, pp. 58–63. doi:10.1145/3098243.3098267.

[25] M. Nair, T. A. Cappello, S. Dang, M. A. Beach, Rigorous Analysis of Data Orthogonalization for Self-Organizing Maps in Machine Learning Cyber Intrusion Detection for LoRa Sensors, IEEE Transactions on Microwave Theory and Techniques 71 (1) (2023) 389–408. doi:10.1109/TMTT.2022.3223122.
URL https://ieeexplore.ieee.org/document/9965952/

[26] H. Aksu, A. S. Uluagac, E. S. Bentley, Identification of Wearable Devices with Bluetooth, IEEE Transactions on Sustainable Computing 6 (2) (2021) 221–230. doi:10.1109/TSUSC.2018.2808455.

[27] M. Shen, M. Wei, L. Zhu, M. Wang, Classification of Encrypted Traffic With Second-Order Markov Chains and Application Attribute Bigrams, IEEE Transactions on Information Forensics and Security 12 (8) (2017) 1830–1843. doi:10.1109/TIFS.2017.2692682.

[28] W. Pan, G. Cheng, Y. Tang, Wenc: Https encrypted traffic classification using weighted ensemble learning and markov chain, in: 2017 IEEE Trustcom/BigDataSE/ICESS, IEEE, 2017, pp. 50–57.

[29] H. Noura, T. Hatoum, O. Salman, J.-P. Yaacoub, A. Chehab, LoRaWAN security survey: Issues, threats and possible mitigation techniques, Internet of Things 12 (2020) 100303. doi:10.1016/j.iot.2020.100303.

[30] K.-H. Lam, C.-C. Cheung, W.-C. Lee, RSSI-Based LoRa Localization Systems for Large-Scale Indoor and Outdoor Environments, IEEE Transactions on Vehicular Technology 68 (12) (2019) 11778–11791, conference Name: IEEE Transactions on Vehicular Technology. doi:10.1109/TVT.2019.2940272.

[31] A. Acar, H. Fereidooni, T. Abera, A. K. Sikder, M. Miettinen, H. Aksu, M. Conti, A.-R. Sadeghi, S. Uluagac, Peek-a-boo: i see your smart home activities, even encrypted!, in: Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks, WiSec '20, Association for Computing Machinery, New York, NY, USA, 2020, pp. 207–218. doi:10.1145/3395351.3399421.
URL https://doi.org/10.1145/3395351.3399421

[32] J. Martin, T. Mayberry, C. Donahue, L. Foppe, L. Brown, C. Riggins, E. C. Rye, D. Brown, A Study of MAC Address Randomization in Mobile Devices and When it Fails, Proceedings on Privacy Enhancing Technologies 2017 (4) (2017) 365–383. doi:10.1515/popets-2017-0054.
URL http://content.sciendo.com/view/journals/popets/2017/4/article-p365.xml

[33] G. Celosia, M. Cunche, Saving Private Addresses: An Analysis of Privacy Issues in the Bluetooth-Low-Energy Advertising Mechanism, in: MobiQuitous 2019 - 16th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services, 2019, pp. 1–10, core B, Qualis B1. doi:10.1145/3360774.3360777.
URL https://hal.inria.fr/hal-02394629

[34] S. Pélissier, J. Aalmoes, A. K. Mishra, M. Cunche, V. Roca, D. Donsez, Privacy-preserving pseudonyms for LoRaWAN, in: 17th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec 2024), 2024.

[35] P. Spadaccino, F. G. Crinó, F. Cuomo, LoRaWAN Behaviour Analysis through Dataset Traffic Investigation, Sensors 22 (7) (2022) 2470, number: 7 Publisher: Multidisciplinary Digital Publishing Institute. doi:10.3390/s22072470.
URL https://www.mdpi.com/1424-8220/22/7/2470

[36] M. M. R. Monjur, J. Heacock, R. Sun, Q. Yu, An Attack Analysis Framework for LoRaWAN applied Advanced Manufacturing, in: 2021 IEEE International Symposium on Technologies for Homeland Security (HST), 2021, pp. 1–7. doi:10.1109/HST53381.2021.9619847.

[37] F. Hessel, L. Almon, F. Álvarez, ChirpOTLE: A Framework for Practical LoRaWAN Security Evaluation, in: Proceedings of the 13th ACM Conference on Security and Privacy in Wireless and Mobile Networks, 2020, pp. 306–316, arXiv:2005.11555 [cs]. doi:10.1145/3395351.3399423.

URL http://arxiv.org/abs/2005.11555
[38] C. Arackaparambil, S. Bratus, A. Shubina, D. Kotz, On the reliability of wireless fingerprinting using clock skews, in: Proceedings of the third ACM conference on Wireless network security - WiSec '10, ACM Press, Hoboken, New Jersey, USA, 2010, p. 169. doi:10.1145/1741866.1741894.
URL http://portal.acm.org/citation.cfm?doid=1741866.1741894
[39] J. Hua, H. Sun, Z. Shen, Z. Qian, S. Zhong, Accurate and Efficient Wireless Device Fingerprinting Using Channel State Information, in: IEEE INFOCOM 2018 - IEEE Conference on Computer Communications, IEEE, Honolulu, HI, 2018, pp. 1700–1708. doi:10.1109/INFOCOM.2018.8485917.
URL https://ieeexplore.ieee.org/document/8485917/
[40] M. Arazzi, S. Nicolazzo, A. Nocera, A novel IoT trust model leveraging fully distributed behavioral fingerprinting and secure delegation, Pervasive and Mobile Computing 99 (2024) 101889. doi:10.1016/j.pmcj.2024.101889.
[41] M. Aernouts, R. Berkvens, K. Van Vlaenderen, M. Weyn, Sigfox and LoRaWAN Datasets for Fingerprint Localization in Large Urban and Rural Areas (Jul. 2019). doi:10.5281/zenodo.3904158.
[42] K. N. Choi, H. Kolamunna, A. Uyanwatta, K. Thilakarathna, S. Seneviratne, R. Holz, M. Hassan, A. Y. Zomaya, LoRadar: LoRa sensor network monitoring through passive packet sniffing, SIGCOMM Comput. Commun. Rev. 50 (4) (2020) 10–24. doi:10.1145/3431832.3431835.
[43] L. Bhatia, M. Breza, R. Marfievici, J. A. McCann, LoED: The LoRaWAN at the Edge Dataset (Sep. 2020). doi:10.5281/zenodo.4121430.
[44] A. Povalac, J. Kral, LoRaWAN Traffic Analysis Dataset (Jun. 2023). doi:10.5281/zenodo.8090619.
[45] R. Overdorf, M. Juarez, G. Acar, R. Greenstadt, C. Diaz, How Unique is Your .onion?: An Analysis of the Fingerprintability of Tor Onion Services, in: Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, ACM, Dallas Texas USA, 2017, pp. 2021–2036. doi:10.1145/3133956.3134005.
[46] M. Aernouts, R. Berkvens, K. Van Vlaenderen, M. Weyn, Sigfox and LoRaWAN Datasets for Fingerprint Localization in Large Urban and Rural Areas, Data 3 (2) (2018) 13. doi:10.3390/data3020013.
[47] S. M. Nguyen, D. V. Le, P. J. M. Havinga, Seeing the world from its words: All-embracing Transformers for fingerprint-based indoor localization, Pervasive and Mobile Computing 100 (2024) 101912. doi:10.1016/j.pmcj.2024.101912.
[48] L. Babun, H. Aksu, L. Ryan, K. Akkaya, E. S. Bentley, A. S. Uluagac, Z-IoT: Passive Device-class Fingerprinting of ZigBee and Z-Wave IoT Devices, in: ICC 2020 - 2020 IEEE International Conference on Communications (ICC), 2020, pp. 1–7. doi:10.1109/ICC40277.2020.9149285.
[49] A. S. Lutakamale, H. C. Myburgh, A. de Freitas, RSSI-based fingerprint localization in LoRaWAN networks using CNNs with squeeze and excitation blocks, Ad Hoc Networks 159 (2024) 103486. doi:10.1016/j.adhoc.2024.103486.
[50] C. Neumann, O. Heen, S. Onno, An empirical study of passive 802.11 Device Fingerprinting, arXiv:1404.6457 [cs] (Apr. 2014). arXiv:1404.6457.
[51] R. W. Klein, M. A. Temple, M. J. Mendenhall, Application of wavelet-based RF fingerprinting to enhance wireless network security, Journal of Communications and Networks 11 (6) (2009) 544–555. doi:10.1109/JCN.2009.6388408.
[52] M. Korczyński, A. Duda, Markov chain fingerprinting to classify encrypted traffic, in: IEEE INFOCOM 2014 - IEEE Conference on Computer Communications, 2014, pp. 781–789. doi:10.1109/INFOCOM.2014.6848005.
[53] L. Alliance, RP2-1.0.3 LoRaWAN® Regional Parameters, Tech. rep., LoRa Alliance (2021).
[54] A. Abdelghany, B. Uguen, C. Moy, D. Lemur, On Superior Reliability of Effective Signal Power versus RSSI in LoRaWAN, in: 2021 28th International Conference on Telecommunications (ICT), IEEE, London, United Kingdom, 2021, pp. 1–5. doi:10.1109/ICT52184.2021.9511510.
[55] M. Anjum, M. A. Khan, S. A. Hassan, A. Mahmood, M. Gidlund, Analysis of RSSI fingerprinting in LoRa networks, 2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC) (2019) 1178–1183.
[56] S. Yadav, S. Shukla, Analysis of k-fold cross-validation over hold-out validation on colossal datasets for quality classification, in: 2016 IEEE 6th International Conference on Advanced Computing (IACC), IEEE, 2016, pp. 78–83.
[57] D. Arp, E. Quiring, F. Pendlebury, A. Warnecke, F. Pierazzi, C. Wressnegger, L. Cavallaro, K. Rieck, Dos and Don'ts of Machine Learning in Computer Security, in: 31st USENIX Security Symposium (USENIX Security 22), 2022, pp. 3971–3988.
[58] G. E. Batista, A. L. Bazzan, M. C. Monard, Balancing Training Data for Automated Annotation of Keywords: A Case Study., in: WOB, 2003, pp. 10–18.
[59] N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, SMOTE: Synthetic minority over-sampling technique, Journal of artificial intelligence research 16 (2002) 321–357.
[60] G. Lemaître, F. Nogueira, C. K. Aridas, Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning, Journal of Machine Learning Research 18 (17) (2017) 1–5.
URL http://jmlr.org/papers/v18/16-365.html
[61] S. Kaufman, S. Rosset, C. Perlich, O. Stitelman, Leakage in data mining: Formulation, detection, and avoidance, ACM Transactions on Knowledge Discovery from Data (TKDD) 6 (4) (2012) 1–21.
[62] A. K. Mishra, A. C. Viana, N. Achir, Introducing benchmarks for evaluating user-privacy vulnerability in WiFi, in: 2023 IEEE 97th Vehicular Technology Conference (VTC2023-Spring), IEEE, 2023, pp. 1–7.
[63] N. Li, N. Zhang, S. K. Das, B. Thuraisingham, Privacy preservation in wireless sensor networks: A state-of-the-art survey, Ad Hoc Networks 7 (8) (2009) 1501–1514. doi:10.1016/j.adhoc.2009.04.009.
[64] B. V. Dos Santos, A. Vergütz, R. T. Macedo, M. Nogueira, A dynamic method to protect user privacy against traffic-based attacks on smart home, Ad Hoc Networks 149 (2023) 103226.
[65] N. L. Giménez, J. M. Solé, F. Freitag, Embedded federated learning over a LoRa mesh network, Pervasive and Mobile Computing 93 (2023) 101819.
[66] C. Dwork, K. Kenthapadi, F. McSherry, I. Mironov, M. Naor, Our data, ourselves: Privacy via distributed noise generation, in: Advances in Cryptology-EUROCRYPT 2006: 24th Annual International Conference on the Theory and Applications of Cryptographic Techniques, St.

Petersburg, Russia, May 28-June 1, 2006. Proceedings 25, Springer, 2006, pp. 486–503.

[67] M. Bor, U. Roedig, LoRa Transmission Parameter Selection, in: 2017 13th International Conference on Distributed Computing in Sensor Systems (DCOSS), IEEE, Ottawa, ON, 2017, pp. 27–34. doi:10.1109/DCOSS.2017.10.
URL http://ieeexplore.ieee.org/document/8271941/

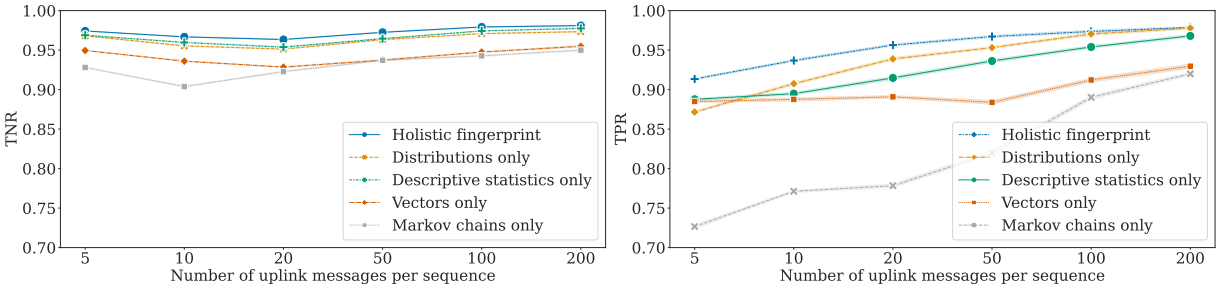
[68] A. Lavric, V. Popa, A LoRaWAN: Long range wide area networks study, in: 2017 International Conference on Electromechanical and Power Systems (SIELMEN), 2017, pp. 417–420. doi:10.1109/SIELMEN.2017.8123360.

[69] E. Bäumker, A. Miguel Garcia, P. Woias, Minimizing power consumption of LoRa ® and LoRaWAN for low-power wireless sensor nodes, Journal of Physics: Conference Series 1407 (1) (2019) 012092. doi:10.1088/1742-6596/1407/1/012092.

[70] J. Lopez, R. Rios, F. Bao, G. Wang, Evolving privacy: From sensors to the Internet of Things, Future Generation Computer Systems 75 (2017) 46–57. doi:10.1016/j.future.2017.04.045.

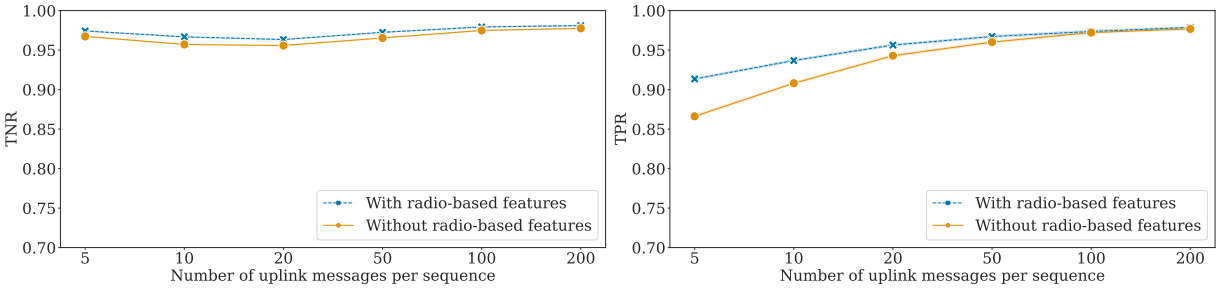
[71] V. Srinivasan, J. Stankovic, K. Whitehouse, Protecting your daily in-home activity information from a wireless snooping attack, in: Proceedings of the 10th International Conference on Ubiquitous Computing, ACM, Seoul Korea, 2008, pp. 202–211. doi:10.1145/1409635.1409663.

[72] D. Balouek, A. Carpen Amarie, G. Charrier, F. Desprez, E. Jeannot, E. Jeanvoine, A. Lèbre, D. Margery, N. Niclausse, L. Nussbaum, O. Richard, C. Pérez, F. Quesnel, C. Rohr, L. Sarzyniec, Adding virtualization capabilities to the Grid'5000 testbed, in: I. I. Ivanov, M. van Sinderen, F. Leymann, T. Shan (Eds.), Cloud Computing and Services Science, Vol. 367 of Communications in Computer and Information Science, Springer International Publishing, 2013, pp. 3–20. doi:10.1007/978-3-319-04519-1_1.



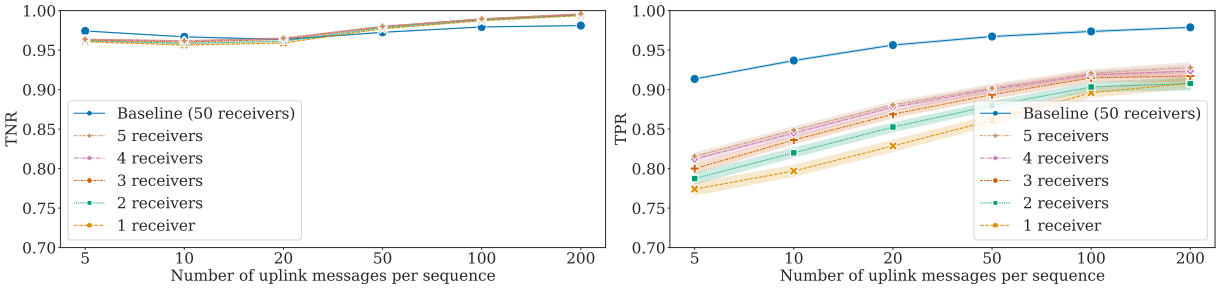
(a) TNR. (b) TPR.

Figure A.15: Performance w.r.t. fingerprint representations.



(a) TNR. (b) TPR.

Figure A.16: Performance w.r.t. radio-based features.



(a) TNR. (b) TPR.

Figure A.17: Performance w.r.t. controlled listening stations.

Appendix A. Complementing balanced accuracy

For completeness, we provide TNR and TPR values corresponding to Figures in the main body. More precisely, Figures A.15a and A.15b correspond to Figures 9 ; A.16a and A.16b to 12 ; A.17a and A.17b to 13. We observe similar trends as discussed in Section 9.3, with a relatively stable TNR and an increasing TNR based on sequence lengths. No setting reaches as low values as for time-based values only. However, Figure A.15b shows that Markov chain require at least 100 messages per sequence to achieve a TPR comparable to other fingerprinting methods. This is expected, given that their statistical nature relies on sufficient sequence data to accurately model state transitions.