Device re-identification in LoRaWAN through messages linkage

Samuel Pélissier, Mathieu Cunche, Vincent Roca, Didier Donsez

Device re-identification in LoRaWAN through messages linkage

Samuel Pélissier
University of Lyon, INSA-Lyon, Inria, CITI Lab.
Lyon, France
samuel.pelissier@insa-lyon.fr

Mathieu Cunche
University of Lyon, INSA-Lyon, Inria, CITI Lab.
Lyon, France
mathieu.cunche@insa-lyon.fr

Vincent Roca
University Grenoble Alpes, Inria
Grenoble, France
vincent.roca@inria.fr

Didier Donsez
University Grenoble Alpes, LIG
Grenoble, France
didier.donsez@univ-grenoble-alpes.fr

ABSTRACT

In LoRaWAN networks, devices are identified by two identifiers: a globally unique and stable one called `DevEUI`, and an ephemeral and randomly assigned pseudonym called `DevAddr`. The association between those identifiers is only known by the network and join servers, and is not available to a passive eavesdropper.

In this work, we consider the problem of linking the `DevAddr` with the corresponding `DevEUI` based on passive observation of the LoRa traffic transmitted over the air. Leveraging metadata exposed in LoRa frames, we devise a technique to link two messages containing respectively the `DevEUI` and the `DevAddr`, thus identifying the link between those identifiers. The approach is based on machine learning algorithms using various pieces of information including timing, signal strength, and fields of the frames. Based on an evaluation using a real-world dataset of 11 million messages, with ground truth available, we show that multiple machine learning models are able to reliably link those identifiers. The best of them achieves an impressive true positive rate of over 0.8 and a false positive rate of 0.001.

CCS CONCEPTS

• **Security and privacy** → **Privacy protections**; • **Computer systems organization** → **Sensor networks**; • **Networks** → *Link-layer protocols.*

KEYWORDS

LoRaWan, re-identification attack, IoT, privacy

ACM Reference Format:
Samuel Pélissier, Mathieu Cunche, Vincent Roca, and Didier Donsez. 2022. Device re-identification in LoRaWAN through messages linkage. In *Proceedings of the 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '22), May 16–19, 2022, San Antonio, TX, USA.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3507657.3528556

1 INTRODUCTION

LoRaWAN is a technology allowing for long-range and low bandwidth communications. It is particularly suited for devices with low energy resources such as sensors running on battery for extended period of time. It is used in domain such as health, well-being, and smart city [11]. As of today, it is estimated that there are hundreds of millions of devices[1] and 166 LoRaWAN network operators[2] deployed worldwide.

In LoRaWAN, end-devices (e.g., sensors) are identified by two elements, the `DevEUI` and the `DevAddr`. The `DevEUI` is a globally unique identifier (similar to a MAC address) tied to the device but only exposed in a single message during the join procedure. The `DevAddr` is an identifier assigned to the device by the network as part of the join procedure. After the join procedure, only the `DevAddr` is used and the `DevEUI` is never observed again. As part of security features of LoRaWAN, the assignment of the `DevAddr` is done via an encrypted message, thus the association between a `DevEUI` and a `DevAddr` is never exposed in cleartext over the air. The `DevAddr` can be renewed, via `rejoin-request` [7], to a new random value. The various `DevAddr` assumed by a device are thus equivalent to unlinkable pseudonyms from the point of view of an external observer.

Linking multiple identifiers used by a wireless device is a problem that has been mainly studied in Wi-Fi [23] and Bluetooth/BLE [5, 15] where address randomization is becoming the norm. For LoRaWAN, we conducted a preliminary work linking `DevAddr` with `DevEUI` by leveraging the timing closeness between the associated messages [2].

In the present work, we propose a consolidated method to find the association between a `DevEUI` and a `DevAddr`. After a passive observation of the LoRa traffic over the air, similarly to [2], the `DevEUI`/ `DevAddr` association is obtained by linking `join-request` messages with the following `uplink` messages, this time leveraging a large set of metadata such as relative timing, signal reception indicators (RSSI, SNR, etc.), and values included directly inside the headers.

The contributions of this paper are the following:

- We introduce a machine learning approach to re-identify devices by linking `join-request` with `uplink` messages.
- We identify pieces of information found in LoRaWAN traffic that can be leveraged for message linking.
- We demonstrate through experimentation on a representative dataset that our approach is able to reliably re-identify devices.

WiSec '22, May 16–19, 2022, San Antonio, TX, USA

ACM ISBN 978-1-4503-9216-7/22/05…$15.00
https://doi.org/10.1145/3507657.3528556

[1] https://www.semtech.com/lora
[2] https://lora-alliance.org/

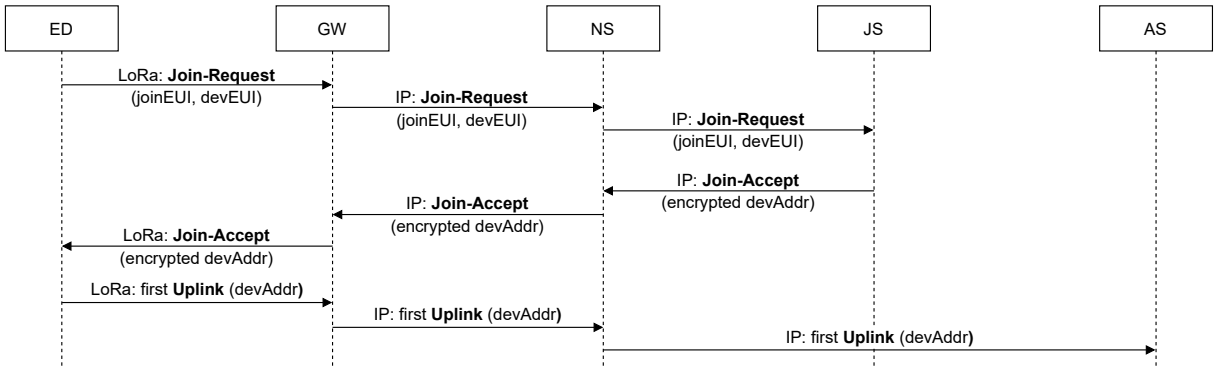

Figure 1: Join procedure using OTAA

- We discuss several countermeasures that could be deployed to thwart our attack.

2 BACKGROUND

Standardized by the LoRa Alliance, LoRaWAN is an open source protocol [7] used as a MAC layer to connect end-devices to servers though gateways (GW). The physical link between the end-device and GW leverages LoRa, a proprietary radio frequency modulation. Thus, the role of the gateway is to translate the LoRa traffic to IP and forward it to or from the server. More precisely, there are three types of servers:

- The Join Server (JS) is responsible of the enrollment of the end-device into the network;
- The Network Server (NS) routes the traffic to the relevant Application Server;
- The Application Server (AS) holds the actual application's logic.

In numerous implementations, the NS, AS and JS are all located on the same node, but they can be physically separated. In any case, a message going from the end-device to a server is called `uplink` and, in the opposite direction, `downlink`.

2.1 Addressing

Each end-device owns a unique IEEE EUI64 identifier named `DevEUI`, provided by the manufacturer and constant over its lifetime. The first 3 bytes contain the OUI of the end-device (Organizationally Unique Identifier, registered to the IEEE).

Apart from this static address, LoRaWAN also uses a new dynamic `DevAddr` every time an end-device is activated or re-joins the network. Generated by the NS by concatenating the variable length network identifier (7 to 24 bits) and random data, this 32 bits identifier is not bound to a unique device. Multiple end-devices can theoretically use the same `DevAddr` at the same time. Thus, the JS keeps track of the association between `DevEUI` and tuples containing the `DevAddr` as well as a the network's session key.

2.2 End-Device activation

A newly deployed end-device must follow a linking procedure to be able to communicate with the NS. Two methods exist: Over-the-Air Activation (OTAA) and Activation By Personalization (ABP). They mainly differ in where cryptographic information is stored: in ABP, the various keys as well as the `DevAddr` are saved directly on the end-device before the activation, whereas OTAA derives them during the join procedure.

In this work, we only consider OTAA, the standard procedure (see figure 1). The end-device first sends a `join-request` message containing notably the `DevEUI` and `JoinEUI`, an IEEE EUI64 unique identifier corresponding to the targeted JS. This is the only time the `DevEUI` is available unencrypted on the air.

Then, the relevant JS answers with a `join-accept` message, providing additional information to finalize the generation of the keys required to setup a secure session with the AS, as well as a `DevAddr`. However the `join-accept` is encrypted, and the following `uplink` messages are the only ones that provide the `DevAddr` in clear text. This mechanism is meant to avoid linking the unique identifier of the end-device, the `DevEUI`, with the traffic containing its ephemeral identifier, the `DevAddr`.

3 MOTIVATIONS AND THREAT MODEL

Being able to link the `DevAddr` with the `DevEUI` of an end-device, known as a re-identification attack, can be leveraged to threaten the security and privacy of LoRaWAN networks. And since join and rejoin procedures are similar [7], it is also possible to link the permanent `DevEUI` with *multiple* `DevAddr` following the various `rejoin-request`.

From the security viewpoint, which is not the main objective of this work, this attack can be exploited to target specific devices with reactive and selective jamming [4, 18].

From the privacy viewpoint, since the notion of privacy varies through context and time [9], reducing it to basic questions is beneficial. In our context, privacy applies to three categories of information: identity (*who?*), activity (*what/when?*) and location (*where?*) [10]. By design, LoRaWAN separates the identity (via the `DevEUI`) from the activity (via the `DevAddr`). Linking them provides more accurate information about the end-device and its potential applications, both exploitable by an attacker to map it with an already known identity, activity or location. For example, parking lot end-device data could help inferring the activity of a household [14], leading to a privacy leakage that involves both activity and location. Another example, the OUI part of the `DevEUI` can be used

to infer the manufacturer, or even the end-device model [16], and it can help mapping a type of end-device with a company or person.

In any case, eavesdropping LoRaWAN traffic is straightforward. Any attacker knowing which band to listen to can eavesdrop the communication [22]. To do so, one can use a cheap LoRa device to listen on a specific channel [19] or a more expensive but still affordable gateway to monitor multiple channels[3].

Then, LoRa benefits from a wide transmission range by design, meaning that the attack does not require a physical proximity but can be conducted from kilometers away. This is a major difference with eavesdropping attacks in wireless protocols such as Bluetooth or WiFi [3, 8]. It also means that the eavesdropping device can listen for long periods of time without being detected, and can thus collect a significant amount of messages.

Threat model: Our threat model relies on several assumptions. First, we assume that the encryption layer is robust: it is impossible to access the content of encrypted payloads. Then, we assume a passive attacker: the traffic is left untouched and only listened to, without injecting or altering messages. This is a reasonable assumption as the messages benefit from an integrity protection. Likewise, we assume the end-device is left unmodified, and the attacker cannot physically access it. Finally, we assume that the attacker controls several receivers in the targeted area, silently, potentially over extended periods of time. As commercial gateways cost a few hundreds euros (see above), achieving good coverage in a citywide area is realistic.

4 LINKING JOIN REQUESTS AND UPLINK MESSAGES

In this section, we present the process and the various pieces of information used to link the `join-request`, containing a `DevEUI` and the following first `uplink` message. In machine learning, such pieces of information are called *features*.

Two specificities can be leveraged for this matching. The first one is inherent to the protocol itself: a `join-request` is generally closely followed by an `uplink` with a fresh `DevAddr`. There is no obligation for the end-device to use the newly acquired `DevAddr` right after receiving it; however, it makes sense in general for a sensor either to ask for its configuration or to start sending data as soon as possible.

The second one is the fact that each end-device has a form of fingerprint relative to a group of gateways: its distance to them and the terrain in between both affect the radio features. Although LoRa end-devices can be mobile, many of them are static (e.g., a water metering sensors). Likewise, the emitted signal of an end-device remains constant through time: its hardware is built to broadcast data with a specific power and should not significantly vary through time. Thus, two messages coming from the same static end-device should have physical similarities when received by a given gateway.

Based on these assumptions, we analyze the variation of features among messages in our dataset by separating known first `uplink` messages from other `uplink` messages. The distribution of the data according to each feature is then analyzed to spot differences between the two sets. If the discrepancies are important enough, the studied feature can be used to distinguish valid from invalid links. A complete list is available in table 1.

The most basic form of feature is directly extracted from the unencrypted header of the message. For example, each `uplink` message contains a 2-bytes frame counter (`FCnt`), starting at zero following a join or re-join procedure and incremented by 1 with each subsequent `uplink`.

Alternatively, it is possible to extract more complex features based on the transmission itself. As LoRa frames are broadcast, they can be received by multiple gateways. This is not a problem when working with header fields such as the frame counter. However, physics-based features vary from gateway to gateway and cannot easily be converted to a single value. Thus, we use various vectors, with each index corresponding to a gateway from the dataset. When a message is received by a gateway, the relevant value is placed at the corresponding index and it is then possible to compute various distances. For example, by setting a 1 if a gateway received the message and a 0 if it did not, we can compute a vector coding the message reception patterns by gateways. Then, we compute the euclidean distance ($\texttt{GW}_{\texttt{dist}}$) between the vector of a `join-request` and the ones of `uplink` messages. Another example, for a join request j and an uplink u received by n gateways, the ESP distance is computed as follows:

$$\texttt{ESP}_{\texttt{dist}}(j,u) = \sqrt{(j_{ESP1}-u_{ESP1})^2+(j_{ESP2}-u_{ESP2})^2+\ldots+(j_{ESPn}-u_{ESPn})^2)}$$

The closest the vectors are, the most likely the same end-device sent both the `join-request` and the corresponding `uplink`. This process is followed for all features with the *dist* suffix.

Name	Description
`FCnt`	Frame counter
`plen`	uplink payload length
`OUI`	OUI extracted from the DevEUI
`Datarate`	Datarate
`SF`	Spreading Factor
$\texttt{ts}_{\texttt{diff}}$	Time of arrival difference between the `join-request` and the studied `uplink`
$\texttt{ESP}_{\texttt{dist}}$	Estimated Signal Power euclidean distance
$\texttt{RSSI}_{\texttt{dist}}$	Received Signal Strength Indication euclidean distance
$\texttt{SNR}_{\texttt{dist}}$	Signal to Noise Ratio euclidean distance
$\texttt{ts}_{\texttt{dist}}$	Timestamps euclidean distance
$\texttt{DevAddr}_{\texttt{diff}}$	Time of arrival difference between two `uplink` messages with identical `DevAddr`
$\texttt{GW}_{\texttt{dist}}$	Euclidean distance based on gateways receiving the messages

Table 1: Features used by the linking process

[3]For example, The Things Gateway costs 300€: https://www.thethingsnetwork.org/docs/gateways/gateway/. A smaller version is even cheaper, costing 70€ https://www.thethingsnetwork.org/docs/gateways/thethingsindoor/

5 METHODOLOGY

In this section, we consider a machine learning approach to link `join-request` with `uplink` messages, and therefore `DevEUI` with `DevAddr`, based on the features presented in section 4 (see figure 2). The process can be seen as a binary classification problem, as we need to distinguish pairs of messages that are actually linked from pairs that are not.

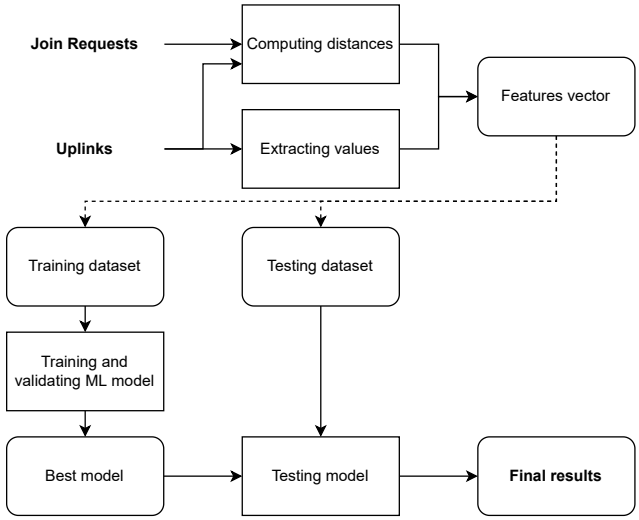

Figure 2: Overview of the methodology

Dataset: To conduct our experiments, we use a real-world dataset including 740 000 `join-request` and 10 570 000 `uplink` messages of LoRaWAN 1.0. They were received by a set of a hundred gateways deployed in various places in the city of Grenoble (France), listening to the EU 868MHz band, from the 23th June 2020 to the 3rd January 2022.[4]

As we control the LoRaWAN network server, we have access to the content of the payloads as well as the ground truth for links between 211 known `DevEUI` and 1024 known `DevAddr` identifiers. This network is operated by the university and the end-devices are managed by a community of researchers for experimental purposes. Although the end-devices are used in research settings, their implementation and deployment aim at recreating a realistic real-world LoRaWAN network.

Performance metrics: In order to evaluate the classification efficiency, we use the following metrics:

- True positive (TP): the link found is valid;
- False positive (FP): the link found is invalid;
- True negative (TN): no link was found, correctly;
- False negative (FN): no link was found, erroneously;
- True Positive Rate (TPR);
- False Positive Rate (FPR);
- Matthew correlation coefficient (MCC)[5].

Data preparation: The machine learning models are built with two sets: a set of `join-request` messages $\mathcal{J}$ and another of `uplink` messages $\mathcal{U}$. Note we only deal with the ground truth: messages received from other operators were previously removed. More precisely, a subset of known links $\mathcal{K}$ is used to detect the unique valid pairs $(j, u_v) \in \mathcal{K}$, representing the first class. The second class is built by using the randomly selected leftover `uplink` messages to create invalid pairs $(j, u_i) \notin \mathcal{K}$. This ensures the second class is heterogeneous and contains a vast variety of associations.

Both types of pairs are converted into vectors v containing a list of features f such as $v(j, u) = f_1, f_2, ..., f_n$ with the last value f_n being an integer corresponding to the link state between j and u. Thus, f_n is equal to 1 for valid links (j, u_v) and 0 for invalid links (j, u_i).

Considered classifiers: Following the works of Acar et al. [1], we choose to study seven classifiers (CLF): Decision Tree (DT), Naive Bayes (NB), Logistic regression (LR), K-Nearest Neighbours (kNN), Random Forest (LR), AdaBoost (AB), and LightBGM (LBGM).

We replaced XGBoost [1] by LightBGM, a faster alternative for gradient boosting. Instead of sorted continuous values, it leverages histograms to build the internal trees, reducing the number of splitting points to analyse [12].

Training: Once the data has been transformed into the expected format, we use the scikit-learn Python library [6], which provides many algorithms [17]. In our work, we consider several of them with the goal to identify the most efficient one.

Validations methods: In order to produce robust models against unseen data, the dataset is split into two subsets: 75% is used for training and validation, and the remaining 25% is only exploited during the testing phase. We also use 5-fold cross validation: the dataset is divided into 4 subsets training a model and the last one is used to test it. More precisely, we leverage a stratified k-fold cross validation to preserve the proportion of each class inside the subsets. Using such method improves results, minimizing the bias and variance [13] at the cost of higher computing resources [24]. It also proves an attacker does not need to listen for communications for years before being able to link the traffic. This process is repeated 5 times to further reduce the bias and variance [20].

6 EXPERIMENTAL RESULTS

In this section, we present the evaluation of the machine learning classifiers and data presented in section 5 . We select a subset that ranges from September 1st, to December 31st, 2021. This equals to 1086 valid links and 165 534 invalid links.

The frame counter is an obvious indicator to detect the first `uplink` message, and obfuscating it is a potential mitigating solution for the re-identification attack (see section 7). So, in order to further assess the importance of this feature for the success of the attacks, we considered two versions of the dataset: one including it, and one without.

[4] More precisely, only the 868.1, 868.3, 868.5, 868.3, 867.1, 867.3, 867.5, 867.7 and 867.9MHz frequencies for `uplink` messages as well as 869.525MHz for downlinks are listened to. The carrier-grade frequencies used by specific operators are not received by the gateways. Thus, some portions of a communication can be lost because the end-device changes to a frequency not monitored. This does not affect our experiments as it is only relevant to traffic coming from uncontrolled operators.

[5] As the two classes are imbalanced, we specifically use the Matthew correlation coefficient (MCC) to compare models. Contrary to accuracy or F1-score, it correctly leverages all data from the confusion matrix and provides a score taking into account both the positive and negative cases [6].

[6] https://scikit-learn.org/stable/index.html (version 1.0.2)

Stability against unseen data: The MCC remains stable for models across the 5-fold cross validation and testing phases[7]. More detailed results are available as artefacts (see section 10).

Performances with the frame counter: A comparison between the various classifiers in table 2 shows that Random Forest presents the best results while including the frame counter in the features, with a true positive rate of 0.7939 and a false positive rate of 0.001. Multiple classifiers show high performances, with Decision Tree, AdaBoost, LightGBM and Random Forest all above a 0.79 MCC value. Thus, an attacker has a high confidence in the links obtained: an overwhelming majority of them is correctly detected and the number of false positives remains low.

CLF	TP	FP	TN	FN	TPR	FPR	MCC
RF	235	42	41317	61	0.7939	0.001	**0.8195**
DT	239	49	41310	57	0.8074	0.0012	0.8173
AB	236	56	41303	60	0.7973	0.0014	0.8013
LGBM	239	68	41291	57	0.8074	0.0016	0.7913
kNN	187	64	41295	109	0.6318	0.0015	0.684
LR	24	33	41326	272	0.0811	0.0008	0.1824
NB	284	10002	31357	12	0.9595	0.2418	0.1398

Table 2: Comparison of classifiers, using the frame counter

CLF	TP	FP	TN	FN	TPR	FPR	MCC
RF	133	29	41330	163	0.4493	0.0007	0.6054
DT	175	115	41244	121	0.5912	0.0028	0.5944
AB	144	70	41289	152	0.4865	0.0017	0.5696
LGBM	191	41	41318	105	0.6453	0.001	**0.7272**
kNN	171	81	41278	125	0.5777	0.002	0.6237
LR	4	1	41358	292	0.0135	0.0	0.1034
NB	33	797	40562	263	0.1115	0.0193	0.0554

Table 3: Comparison of classifiers, without the frame counter

Performances without the frame counter: When the frame counter is ignored (see table 3), most classifiers lose in performance. For example, the true positive rate of the Random Forest classifier decreases from around 0.80 to 0.45. However, for all models, the false positive rate remains close to its previous value and only the number of true positives decreases. Thus, the linking attack can still work with a high confidence on the results, albeit with fewer links found.

Importance of the features: In order to assess the importance of the features, we extracted the weights leveraged by the RF classifier (see table 4). The frame counter is confirmed essential with a weight of 0.3685, followed by the various distances and the payload's length, which amounts for around 13% of the weights without the frame counter[8].

Impact of the number of receivers: Finally, we measure the impact of the number of eavesdropping nodes controlled by the attacker in a specific area. As they are geographically close to each other, they receive a significant subset of messages in common. We selected various sets of three nodes and ran tests with 1, 2, or 3 nodes to compare the impact of the number on the models. Contrary to our intuition, adding new nodes does not necessarily provide better results. Based on our observations, it seems to highly depend on the amount of traffic captured by the node, the higher the better. Exploring further what would improve the global re-identification efficiency is left to future works.

[7]For example, the Random Forest classifier demonstrates a 0.871 average MCC for 5-fold cross validation and a 0.88 during the final test.

[8]We also tested the ACK and ADR flags[7], which seemed promising based on their distributions, but results were inconclusive.

Feature	With FCnt	Without FCnt
FCnt	0.3685	N/A
ts_{diff}	0.115	0.1838
ESP_{dist}	0.0866	0.1284
SNR_{dist}	0.0765	0.1106
$DevAddr_{diff}$	0.0744	0.1178
$RSSI_{dist}$	0.0613	0.0956
ts_{dist}	0.0584	0.0746
plen	0.0546	0.1339
OUI	0.0379	0.0731
Datarate	0.0241	0.0295
SF	0.0237	0.0261
GW_{dist}	0.0191	0.0295

Table 4: Weights of each feature for the Random Forest model

7 COUNTERMEASURES

Since the re-identification attack is effective, as soon as an attacker eavesdrops a `join-request` and the following `uplink` message(s), the question of counter measures naturally arises.

Obfuscating the frame counter: Tests of section 6 highlight the importance of the frame counter, `FCnt`, ranked first in table 4 with a 0.37 coefficient. A first possibility is to hide it, for instance by encrypting a part of the header, or using a random offset instead of starting with value 0 in case of a new end-device joining the network. However, both approaches imply a change of the specifications, although the second one remains easier to implement.

In any case, the benefits would be limited as we also proved that in the absence of the `FCnt`, the attacker can still leverage other features, reaching a reasonable true positive rate of 0.65 (with LBGM) and high confidence on the results. The following question is whether this can be achieved with minimal implications, notably in terms of backward compatibility, without changing the specifications.

Introducing randomness: Introducing randomness in parameters influencing physics-related features, such as ESP, is an option. However, it may not be possible without compromising the correct reception of messages, or greatly modifying the end-devices as well as increasing their energy consumption.

Other possibilities that do not impact LoRaWAN standard are: introducing a random delay before sending the first `uplink` message; padding the payload to hide its actual length; sending multiple first `uplink` messages with false `DevAddr` as decoys.

Obfuscating device identifiers: More disruptive solutions, that require an update of the LoRaWAN standard, in addition to those centered around the `FCnt` feature already discussed, include: sharing a `DevAddr` for multiple end-devices and identifying the devices by their network session key [2]; and using resolvable addresses as in BLE[5].

8 RELATED WORKS

Linking multiple pseudonym identifiers used by devices has been studied in other wireless technologies that use address randomization methods, such as Wi-Fi and Bluetooth-Low-Energy. Methods leveraging the content of the wireless frame as well as their timing have both been presented to correlate frames and identifiers [5, 15, 23].

Physical-layer fingerprinting of LoRa devices has been demonstrated in [19], and could be used to single-out a device. However, this fingerprinting requires specialized hardware (USRP), works with physical-layer and not LoRaWAN itself, and has not been demonstrated in a real-world network.

Privacy considerations for LoRa traffic has been discussed in [14]. More specifically, this work focuses on the inference of information based on traffic metadata, and how it can be obfuscated.

Finally, the linking of `DevAddr` and `DevEUI` in [21] is done by framing the `DevEUI` between two sets of consecutive `DevAddr` with the same behavior. First, their dataset is more homogeneous with 130 end-devices, all deployed for the same application. Then, contrary to our contribution, their method does not take new, unknown end-devices into account and can only link a `DevEUI` effectively framed between two `DevAddr`. Lastly, it only leverages the frame counter as well as the timestamp and obtains a 0.936 accuracy, against 0.9975 for our best model.

9 CONCLUSION

We show it is possible to reliably link two `DevEUI` and `DevAddr`, and thus re-identify devices, by applying machine learning algorithms. Although the frame counter plays a major role in obtaining a 0.8 TPR and 0.001 FPR for the best model, solely obfuscating it is not enough to protect users from this attack, as we obtain a 0.65 TPR and 0.001 FPR by compensating its absence by other pieces of information. We discuss potential countermeasures that require to act on several pieces of information at the same time, with negative consequences for some of them (e.g., backward compliance). Although mitigating the re-identification attack is feasible, it seems there is no easy, perfect solution.

Future works will include fine-tuning the parameters of the models to increase their efficiency, as well as evaluating our approach on other datasets.

10 RESEARCH ARTIFACTS

The (anonymized) data and Python code necessary to build the models and reproduce results are published in a public Git repository: https://gitlab.inria.fr/spelissi/wisec-2022-reproductibility.

ACKNOWLEDGMENTS

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.

REFERENCES

[1] Abbas Acar, Hossein Fereidooni, Tigist Abera, Amit Kumar Sikder, Markus Miettinen, Hidayet Aksu, Mauro Conti, Ahmad-Reza Sadeghi, and Selcuk Uluagac. 2020. 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*. ACM. https://doi.org/10.1145/3395351.3399421

[2] Lucas Ancian and Mathieu Cunche. 2020. *Re-identifying addresses in LoRaWAN networks*. Unpublished Research Report. Inria Rhône-Alpes ; INSA de Lyon. https://hal.inria.fr/hal-02926894

[3] Noah Apthorpe, Danny Yuxing Huang, Dillon Reisman, Arvind Narayanan, and Nick Feamster. 2019. Keeping the Smart Home Private with Smart(er) IoT Traffic Shaping. *Proceedings on Privacy Enhancing Technologies* 2019, 3 (July 2019), 128–148. https://doi.org/10.2478/popets-2019-0040

[4] Emekcan Aras, Nicolas Small, Gowri Sankar Ramachandran, Stéphane Delbruel, Wouter Joosen, and Danny Hughes. 2017. Selective Jamming of LoRaWAN using Commodity Hardware. In *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. ACM, 363–372. https://doi.org/10.1145/3144457.3144478

[5] Johannes K Becker, David Li, and David Starobinski. 2019. Tracking Anonymized Bluetooth Devices. *Proceedings on Privacy Enhancing Technologies* 2019, 3 (July 2019), 50–65. https://doi.org/10.2478/popets-2019-0036

[6] Davide Chicco and Giuseppe Jurman. 2020. The Advantages of the Matthews Correlation Coefficient (MCC) over F1 Score and Accuracy in Binary Classification Evaluation. *BMC Genomics* 21, 1 (Dec. 2020), 6. https://doi.org/10.1186/s12864-019-6413-7

[7] LoRa Alliance Technical Committee. 2017. LoRaWAN® Specification v1.1.

[8] Bogdan Copos, Karl Levitt, Matt Bishop, and Jeff Rowe. 2016. Is Anybody Home? Inferring Activity From Smart Home Network Traffic. In *2016 IEEE Security and Privacy Workshops (SPW)*. IEEE, 245–251. https://doi.org/10.1109/SPW.2016.48

[9] Rachel L. Finn, David Wright, and Michael Friedewald. 2013. Seven Types of Privacy. In *European Data Protection: Coming of Age*. https://doi.org/10.1007/978-94-007-5170-5_1

[10] Dalton A. Hahn, Arslan Munir, and Vahid Behzadan. 2021. Security and Privacy Issues in Intelligent Transportation Systems: Classification and Challenges. *IEEE Intell. Transp. Syst. Mag.* 13, 1 (2021), 181–196.

[11] Jetmir Haxhibeqiri, Eli De Poorter, Ingrid Moerman, and Jeroen Hoebeke. 2018. A Survey of LoRaWAN for IoT: From Technology to Application. *Sensors* 18, 11 (Nov. 2018), 3995. https://doi.org/10.3390/s18113995

[12] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A Highly Efficient Gradient Boosting Decision Tree. *Advances in neural information processing systems* 30 (2017), 3146–3154.

[13] Ron Kohavi. 1995. A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. In *Ijcai*, Vol. 14. 1137–1145.

[14] Patrick Leu, Ivan Puddu, Aanjhan Ranganathan, and Srdjan Čapkun. 2018. 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*. https://doi.org/10.1145/3212480.3212508

[15] Norbert Ludant, Tien D. Vo-Huu, Sashank Narain, and Guevara Noubir. 2021. Linking Bluetooth LE & Classic and Implications for Privacy-Preserving Bluetooth-Based Protocols. In *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 1318–1331. https://doi.org/10.1109/SP40001.2021.00102

[16] Jeremy Martin, Erik Rye, and Robert Beverly. 2016. Decomposition of MAC address structure for granular device inference. In *Proceedings of the 32nd Annual Conference on Computer Security Applications*. ACM, 78–88. https://doi.org/10.1145/2991079.2991098

[17] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.

[18] Toni Perković, Hrvoje Rudeš, Slaven Damjanović, and Antun Nakić. 2021. Low-Cost Implementation of Reactive Jammer on LoRaWAN Network. *Electronics* 10, 7 (April 2021), 864. https://doi.org/10.3390/electronics10070864

[19] Pieter Robyns, Eduard Marin, Wim Lamotte, Peter Quax, Dave Singelée, and Bart Preneel. 2017. 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, 58–63. https://doi.org/10.1145/3098243.3098267

[20] Juan D. Rodriguez, Aritz Perez, and Jose A. Lozano. 2009. Sensitivity Analysis of K-Fold Cross Validation in Prediction Error Estimation. *IEEE transactions on pattern analysis and machine intelligence* 32, 3 (2009), 569–575.

[21] Pietro Spadaccino, Domenico Garlisi, Francesca Cuomo, Giorgio Pillon, and Patrizio Pisani. 2021. Discovery Privacy Threats via Device De-Anonymization in LoRaWAN. In *19th Mediterranean Communication and Computer Networking Conference*. 1–8. https://doi.org/10.1109/MedComNet52149.2021.9501247

[22] Nuno Torres, Pedro Pinto, and Sérgio Ivan Lopes. 2021. Security Vulnerabilities in LPWANs—An Attack Vector Analysis for the IoT Ecosystem. *Applied Sciences* 11, 7 (Jan. 2021), 3176. https://doi.org/10.3390/app11073176

[23] Mathy Vanhoef, Célestin Matte, Mathieu Cunche, Leonardo S. Cardoso, and Frank Piessens. 2016. 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)*. ACM, 413–424. https://doi.org/10.1145/2897845.2897883

[24] Sanjay Yadav and Sanyam Shukla. 2016. 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, 78–83.