Analyzing and explaining privacy risks on time series data: ongoing work and challenges

Tristan Allard (IRISA, Univ. Rennes, Fr), Hira Asghar (LIG, Univ Grenoble-Alpes, Fr), Gildas Avoine (IRISA, INSA Rennes, Fr), Christophe Bobineau (LIG, Univ Grenoble-Alpes, Fr), Pierre Cauchois (Enedis inc., Rennes, Fr), Elisa Fromont (IRISA, Univ. Rennes, IUF, Fr), Anna Monreale (KDD lab, Univ. Pisa, It), Francesca Naretto (KDD lab, Univ. Pisa, It), Roberto Pellungrini (Scuola Normale Superiore, Pisa, It), Francesca Pratesi (CNR, Pisa, It), Marie-Christine Rousset (LIG, Univ Grenoble-Alpes, Fr), Antonin Voyez (Enedis inc., Univ. Rennes, Fr)

ABSTRACT

Currently, privacy risks assessment is mainly performed as audits conducted by data privacy analysts. In the TAILOR project, we promote a more systematic and automatic approach based on interpretable metrics and formal methods to evaluate privacy risks and to control the tension between data privacy and utility. In this paper, we focus on privacy risks raised by publishing time series datasets, and we survey the methods developed in TAILOR to analyze and quantify privacy risks depending on different publisher and attacker models.

1. INTRODUCTION

Mobile devices and smart applications continuously produce a huge amount of data on the behavior of their users over time (e.g., electrical consumption, mobility data). In many domains, collecting and analyzing such data can bring valuable services for end-users, scientists, or decision-makers by providing fine-grained predictions or personalized recommendations. However, time trajectories convey sensitive information which, if analyzed with malicious intent, can lead to a serious violation of the privacy of the individuals involved.

In its simplest form, temporal data are time series that are usually collected in streams (no beginning, no end) which makes them difficult to defend with today's privacy protection methods such as differential privacy (DP) [7]. As a result, time series data are often published after being aggregated. Publishing aggregates (e.g. count, mean or sum of individual values) is a simple and still widely used [11; 10; 12; 8] method for data protection since it allows a data publisher to publish statistics over datasets that remain private.

In this article, we survey ongoing works done in the TAILOR project[1] to detect and explain different types of privacy risks raised by publishing aggregates of time series. TAILOR (Trustworthy AI – Integrating Reasoning, Learning and Optimization) is a European network of research excellence centers working on aspects of trustworthy AI. The presented works are based on a variety of techniques such as machine learning, formal verification, or simulation of privacy attacks.

Providing explanations is crucial for helping data producers to understand the encountered privacy risks so that they can implement an appropriate strategy for mitigating them.

We illustrate the different approaches on a real-world dataset provided by the *Irish Social Science Data Archive (ISSDA) Commission for Energy Regulation (CER)* [2]. This dataset includes time series of electrical consumption of Dublin's households.

The paper is organized as follows. In Section 2, we state the problem of privacy risks assessment that we address. Then, we describe techniques assessing and explaining different types of privacy risks in aggregate time series: re-identification risks (Section 3), membership inference risks (Section 4) and data reconstruction risks (Section 5). Finally, in Section 6 we conclude the paper with some challenges for future work.

2. PROBLEM STATEMENT

After presenting the time series data model that we handle, we describe the problem of privacy risks assessment as a multi-faceted problem depending on the considered publisher and attacker models and also on the desired utility preservation of the published time series dataset.

2.1 Time series data model

We consider univariate time series and we assume that the series in a given dataset are all temporally aligned and recorded with the same frequency. A *time series* is thus a timestamped sequence of scalar values of a given attribute (e.g., the electric consumption recorded at regular intervals of time by an individual smart meter). A *time series dataset* $\mathcal{S}$ is a set of time series in which the values are recorded at the same timestamps in all the time series. We will denote $S_{s,t}$ the value of the time series s at the timestamp t.

In practice, a time series dataset can be stored in different formats and each time series has an identifier to which metadata can possibly be attached.

Within a time series dataset, aggregation functions (i.e., sum, average) can be computed either per timestamp over the values of individual time series grouped into clusters,

[1] https://tailor-network.eu/

[2] https://www.ucd.ie/issda/data/commissionforenergyregulationcer/

or within each individual time series by grouping values by time windows. This results in creating new *aggregate time series* which are likely to present less privacy risks than the original ones.

2.2 The ISSDA dataset

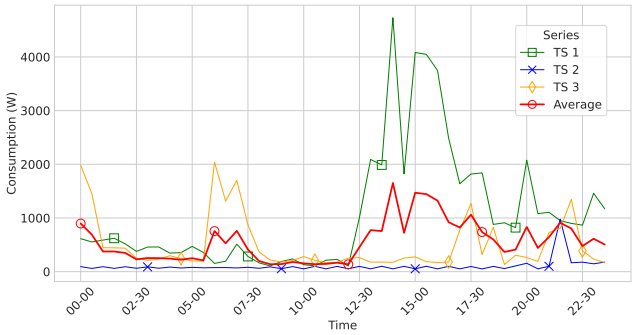


Figure 1: Three illustrative ISSDA series over a single day at 30 min rate and the average aggregate of the three series (in red).

The *CER-ISSDA* dataset mentioned in the introduction is a time series dataset that contains 6435 half-hourly electric consumption time series of Irish individuals collected between July 2009 and December 2010. We removed all series with missing values to obtain a dataset of 4622 full series. Recorded consumption could reach 36 kW, yet 80% of the records are below 1 kW. Figure 1 shows an example of a 24h sample of three ISSDA time series and of the aggregate time series which corresponds to the average of the three series for each considered timestamp.
In addition, metadata are available on customers' demographics, home sizes and equipment associated to the electric consumption time series. We have represented in a uniform way the time series enriched with some of these metadata as a RDF knowledge graph using a simple RDFS ontology[3].

2.3 Publisher model

Given an input private time series dataset, the publisher applies a privacy-preserving data publishing (PPDP) algorithm to output a new time series dataset to be published. In this paper, we consider two types of PPDP algorithms: pseudonymization and aggregation algorithms.
For pseudonymization of time series, we consider that the publisher employs simple techniques consisting in replacing linkable identifiers (such as customer or smart meter numbers for electric consumption time series) by internal identifiers, and in removing the link with personal metadata (such as the name or the address of the customers).
Such simple pseudonymization techniques preserve the time series themselves and thus their full utility for a fine-grained analysis but must be combined with other techniques to reinforce privacy protection.
In our models, the publisher applies aggregation algorithms to the resulting pseudonymized time series datasets.
Two kinds of aggregation are considered for limiting the risk of re-identification of individual time series:

[3]available at `https://raw.githubusercontent.com/fr-anonymous/puck/main/issda\_schema.ttl`

1. Replacing subsets (of a given size) of time series of the original time series dataset by a new time series in which the value at each timestamp is computed as the aggregation of the values at the same timestamp in each subset.

2. Replacing the original timestamps (e.g., every half hour) by new timestamps defining larger time slots (e.g., covering whole days, or half days) for which the associated values are computed as the aggregation of the values corresponding to the original timestamps included in the new time slots.

These two types of aggregation algorithms return aggregated versions of the original time series dataset, the first one with less time series and the same timestamps as in the original dataset, the second one with the same number of time series but less timestamps than the original dataset. In both cases, the *aggregation size* is likely to impact both privacy and utility. In the first case, the *aggregation size* is the number of individual time series in the subset used to compute the aggregation function. In the second case, it is the number of the original timestamps "merged" to define the new time slots. The *series length* in the published time series dataset is also an important parameter. Longer series offer more information to detect specific patterns about an individual time series and are thus particularly susceptible to privacy attacks [25] such as re-identification ones.

2.4 Attacker model

An attacker takes as input a *published time series dataset* and some *background knowledge* to design an algorithm conducting one type of *privacy attacks* based on the background knowledge.
The background knowledge models the partial information known by the attacker, which is of two types:
- Partial knowledge on data: it can be a target individual for which the attacker knows values at certain number of (consecutive) time points; or a target individual time series known by the attacker; or answers to some queries over the original dataset.
- Partial knowledge on the parameters of the PPDP algorithms used by the publisher: it can be the aggregation size or the aggregation function used to produce the published dataset.
The privacy attacks considered in this paper for time series datasets are the following:

1. Re-identification attacks, which succeed if the background knowledge allows to uniquely identify a time series in the published dataset as corresponding to some target individual, thus disclosing the entire time series of that individual.

2. Membership inference attacks, which succeed if they can infer that a target individual time series has been used for computing an aggregate in the published dataset, thus revealing the presence of this individual time series in the original dataset.

3. Data reconstruction attacks, which consist in inferring some data intended to be protected by combining answers to well-chosen queries.

2.5 Automatic privacy risks assessment

Privacy risks assessment is the process of identifying and quantifying the threats raised by possible privacy attacks. Currently, this task is mainly done as audits conducted by data privacy analysts. In the TAILOR project, we promote a more systematic and automatic approach based on interpretable metrics and formal methods to evaluate privacy risks and to control the tension between data privacy and utility. In the remaining of the paper, we survey the different automatic methods that we have developed for time series datasets to analyze and quantify the privacy risks corresponding to the attacker models presented previously.

3. RE-IDENTIFICATION RISKS

Unicity is a widely used measure for evaluating the vulnerability to re-identification risks in tabular personal data, for which k-anonymity has been proposed as a defense in [24]. In Section 3.1, we propose two metrics to define unicity in time series datasets. In Section 3.2, we present an approach developed in the PRUDENCE framework [16], based on a systematic simulation of re-identification attacks based on unicity. Section 3.3 is dedicated to a complementary approach, developed in the EXPERT framework [14], based on a machine learning model for predicting and explaining the privacy risks directly from the time series in input.

3.1 Unicity measure for time series

For tabular data, unicity of a record is defined in function of quasi-identifiers that are attributes for which knowing the values uniquely identify the record in the database. For non tabular data, identifying quasi-identifiers is difficult and thus unicity must be modeled in function of the considered data model. In [26], we have proposed to measure unicity in a time series dataset S as the percentage of series that can be uniquely identified with l consecutive time points. Formally, for a given l, we compute the unicity $u_l^t(S)$ at each time point t as the percentage of times series that are unique in S_l^t, where S_l^t is obtained from S by extracting from each time series the sub-sequence of length l starting at t. Finally, for a given l, we compute the unicity score $U_l(S)$ as either the average or the maximum (depending on the application) of the unicity scores $u_l^t(S)$ over all time points.

In the experiments that we have conducted on the half-hourly ISSDA dataset, we have shown that the unicity score of the whole dataset is, on average over the whole dataset, above 15% for $l = 1$ and above 98% for $l = 3$. This means that few target time series can be uniquely identified with the knowledge of a value at a single time point. Most importantly, this also shows that knowing very few consecutive values makes almost all series of the dataset uniquely identifiable which make time series more vulnerable than classic tabular datasets.

In [15], we have considered an alternative definition of the unicity score for a time series dataset as the percentage of series that can be uniquely identified by the knowledge of l values that are not necessarily consecutive. The computation of this metrics is at the core of the PRUDENCE approach for measuring the risks of re-identification.

3.2 The PRUDENCE approach

In this approach described in [15], we quantify the risk of re-identifying each individual time series in a dataset from knowing l values. For this, we simulate all the possible re-identification attacks in order to select for each individual time series the worst combination of time points for which the values uniquely identify it.For example, for a certain individual the most dangerous combination could be given by the first and second time points, while for another individual it might correspond to third and tenth ones.

More formally, given a time series dataset S, and a parameter l, an individual time series s and a subset $\{t_1, \ldots, t_l\}$ of timestamps, we define as follows the probability $P^S_{\{t_1 \ldots t_l\}}(s)$ of uniquely identifying s in S knowing the background knowledge made of the values S_{s,t_i} at each time point t_i :

$P^S_{\{t_1 \ldots t_l\}}(s) = \frac{1}{|\{s' \in S | \forall i \in [1..l] S_{s',t_i} = S_{s,t_i}\}|}$

Then, we define the risk $Risk_l(s, S)$ of identifying an individual time series s knowing l values as the highest probability $P^S_{\{t_1 \ldots t_l\}}(s)$ over all the possible subsets of time points of size l:

$Risk_l(s, S) = Max\{P^S_{\{t_1 \ldots t_l\}}(s)\}$ where $\{t_1 \ldots t_l\}$ is a subset of l distinct time points.

It models the risk of re-identifying s with the worst attack corresponding to a background knowledge of size l. The computational complexity of calculating $Risk_l(s, S)$ for each time serie s in S may be prohibitive if this parameter l is high and if the number of time points is big since the calculation requires to survey all the possible subsets of size l of the time points in S.

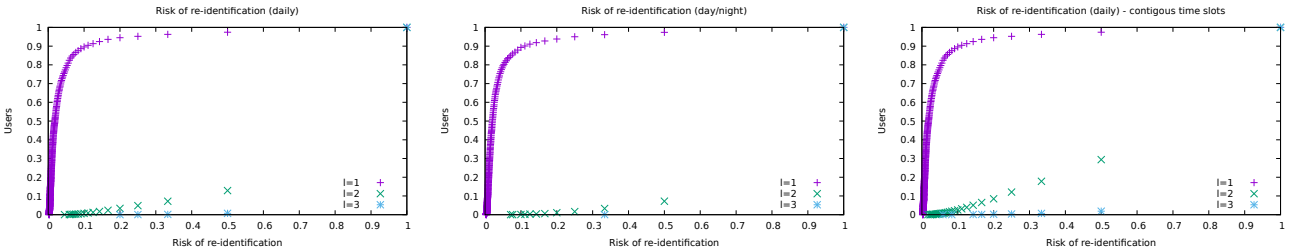
The number of time points in the published dataset depends on the publisher model, more precisely on the chosen aggregation for protecting the published dataset while preserving utility. For instance, for the ISSDA dataset, covering the half-hourly electric consumption of 4,622 individuals over a period of 536 days, we have considered two ways of aggregating the original dataset that vary in the granularity of the time windows grouping the original timestamps. The first publisher model (denoted *daily consumption*) consists in publishing for each day the sum of consumption recorded each half an hour that day. An extract of the corresponding published dataset is given in Figure 2. This aggregated dataset contains 4,622 time series with 536 time points each.

	2009-07-15	2009-07-16	2009-07-17	2009-07-18	2009-07-19	2009-07-20	2009-07-21	2009-07-22	2009-07-23	2009-07-24	...
0	11.198	8.390	7.218	11.322	11.301	2.871	11.589	4.608	12.425	4.936	...
1	6.744	6.945	7.254	7.187	6.802	6.992	12.791	5.772	5.761	6.294	...
2	6.347	8.970	8.793	8.302	10.116	7.827	8.053	5.910	3.843	6.657	...
3	24.175	26.654	32.008	33.025	31.232	25.300	24.409	23.301	32.524	29.789	...
4	50.053	48.807	32.548	46.722	35.204	57.809	53.665	40.277	42.973	42.532	...
...	...	...	...	...	...	...	...	...	...	...	...

Figure 2: Excerpt of the daily consumption published dataset.

The second publisher model (denoted *day/night consumption*) describes each of those 4,622 time series with the double of time points since the aggregation is done by grouping the original timestamps by half days: the published dataset provides the sum of consumption computed over daytime or nighttime hours, for each day in the corresponding period of time. More utility is preserved by this publisher model since the publication of aggregation over smaller time windows allows more fine-grained analysis of the resulting time series, e.g., whether there are significant differences between day and night consumption among specific groups of users.

In order to find a good trade-off between privacy and utility, it is useful to measure and possibly compare the re-



(a) Daily consumption

(b) Day/night time consumption

(c) Daily consumption considering consecutive slots only

Figure 3: Cumulative distribution of re-identification risk for three publisher models of the ISSDA dataset

identifications risks raised by different publisher models for a same attacker model. We have conducted such a comparison of the two above publisher models for the ISSDA dataset, depending on the size l of the background knowledge of an attacker (i.e., the number of time points for which the aggregated consumption values are known).

The results are summarized in Figure 3 (a) and (b) that show the cumulative distribution of the risk $Risk_l(s, S)$ for each time series s in S for l varying from 1 to 3: each point denotes the number of users having *at most* a given value of risk.

For both daily and day/night consumption publisher models, the plots show that the re-identification risk is low when the adversary knows only a single consumption value for their target: 90% of users have a risk of re-identification less than 0.1.

In both models however, the risk increases dramatically with the length of the knowledge of the adversary, even with just two or three known values. In particular, with $l = 3$, we have that 99% of users are re-identifiable with while with $l = 2$, this percentage is around 87% and 93% respectively for the daily consumption publisher model (Figure 3 (a)) and for the day/night consumption publisher model (Figure 3 (b)).

For the daily consumption publisher model, Figure 3 (c) shows that if we consider an attacker model based on knowing values on consecutive slots only, for $l = 2$ the number of users with maximum risk decreases from around 87% to around 70%, and the corresponding curve is generally higher than the one in Figure 3 (a). Similar results are obtained for the day/night consumption model. This shows a dangerous underestimation of the risks if we do not consider the worst combination of time slots for which values are known in the modeling and simulation of re-identification attacks.

3.3 The EXPERT approach

EXPERT [14] is a generic and modular framework for predicting and explaining privacy risks of various type of data by using supervised machine learning techniques and post-hoc explanation methods. In Sections *3.3.1* and *3.3.2*, we describe how we have tailored EXPERT to the prediction and the explanation of re-identification risks of time series and we report on the results on the ISSDA dataset.

3.3.1 Supervised learning the prediction model

The accuracy of the predicting model learned by EXPERT depends on the availability of quality training datasets. We build such training datasets by applying the PRUDENCE approach (Section 3.2) to a sample of the target published time series datasets (ISSDA daily or day/night consumptions). We discretize the training datasets in *high risk* and *low risk*. This is a common practice in privacy risk prediction [15] as the aim of the prediction is to detect individuals that have an important risk of being re-identified. Based on Figure 3 (a) and (b), the resulting training datasets are highly imbalanced. For this reason we focus on learning the re-identification risk for attacker model corresponding to an attacker knowing $l = 2$ values appearing in the time series, which is the case where the risks is the least imbalanced between users (compared to the cases $l = 1$ and $l = 3$). To overcome the imbalance, we exploit the SMOTE oversampling algorithm [4]. For each dataset we split it into training, validation and test set, corresponding respectively to 70%, 20% and 10% of the dataset.

For learning the predicting model we have used Bi-LSTM with the following structure: 2 Bi-LSTM layers (the first of 35 neurons, the second of 20) with recurrent dropout set at 0.30, activation function sigmoid, binary cross entropy as loss and AdaDelta as optimizer[27]. Finally, we set the batch to 64 and we trained the networks for 20 epochs with early stopping, to avoid overfitting, of 3 epochs.

Attacker model	***Publisher model***	***Avg***	***Prec***	***Rec***	***F1***
Gaps	**Daily**	*macro*	0.70	0.70	0.60
		weighted	0.72	0.69	0.70
	Day/ Night	*macro*	0.71	0.73	0.67
		weighted	0.79	0.67	0.68
Cont	**Daily**	*macro*	0.65	0.66	0.65
		weighted	0.69	0.61	0.64
	Day/ Night	*macro*	0.64	0.63	0.64
		weighted	0.69	0.60	0.62

Table 1: EXPERT results (Precision/Recall/F1) on the ISSDA daily and day/night datasets for the "Gaps" and "Continguous" attacker model.

We report our results in Table 1, where prediction performance is given for the daily and day/nigh publisher models depending on the two attacker models considered in Section 3.2 : either the attacker knows 2 values that are not necessarily consecutive (referred to as *Gap*), or 2 consecutive values (referred to as *Contiguous*). The performance are reported both with the *macro* and the *weighted* average. In the first case, the scores are calculated as the mean of all the per-class scores, while in the second case, the weighted mean takes into account the imbalance between classes by weighting each score by the corresponding class support.

The results do not show a high accuracy of the risk prediction. This can be explained by the fact that in the ISSDA daily and day/night datasets, the majority of the series are classified as *high risk*. In such cases where the training data are imbalanced, it is difficult to train the predictor correctly, having to resort to oversampling techniques and to limit overfitting.

3.3.2 *Post-hoc explanations of re-identification risks*

We have used SHAP (SHapley Additive exPlanations) [13], a post-hoc, agnostic method which assigns an importance value to each of the elements in input. SHAP requires some input data on which it can perform the procedure, derived from game theory, for computing the SHAP values by considering each element as a player in a team, and by making the team play with or without it to determine its importance. In our setting, we have exploited the DeepExplainer of SHAP, tailored for deep learning models, which implements a high-speed approximation of SHAP values based on a variant of DeepLIFT[22]. For providing representative input data, we have passed the centroids of a K-means clustering algorithm performed over the training dataset, with $K = 100$.

Two examples of the resulting explanations are presented in Figure 4 for the ISSDA daily and day/night datasets. Each important element for the classification is identified with the position number in the time series (for example `2010-12-27` for the ISSDA daily dataset or `2019-10-10 night` for the ISSDA day/night dataset) with the associated importance value (e.g., `-1.009` or `-0.05217`). The figure shows the most important time slots highlighted in red (respectively in blue) that lead to the prediction of *high risk* (respectively *weak risk*) for the considered time series.

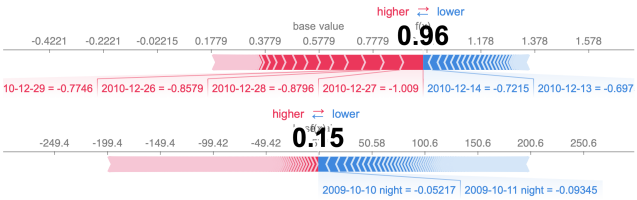


Figure 4: Two examples of SHAP local explanations for the ISSDA daily and day/night datasets.

4. MEMBERSHIP INFERENCE RISKS

Following one of the publisher models described in Section 2.3, we consider an aggregate as being a time series obtained by averaging at each timestamp the values of multiple individual time series.

Our approach consists in choosing a well-adapted machine learning algorithm to predict whether a given target individual time series is likely to be part of a given aggregate and designing a training framework to obtain an accurate attacker model. It is inspired by recent works [20] in the Machine Learning research field that follow the shadow training technique [21] and consist in training an adversarial machine learning model to learn a model of membership inference attack against a target machine learning model. Such learned models are used to infer that a data record is present in the training dataset. A similar shadow training approach is used in [18; 17; 2] to model membership inference attacks against aggregates.

4.1 Methodology

To learn the prediction model of a membership inference attack (MIA), we select a set of target time series in the original time series, and for each of them, we build a balanced training dataset with k aggregate series containing the target series and k aggregate series that do not contain it. This dataset is then used to train a binary classifier that can perform the MIA for the corresponding target. Note that each classifier is specific to the individual target series used to design the training dataset: It can be used only for the same individual but, of course, for different aggregates and for any time period possibly different from the training one.

To evaluate the performance of the attack classifier, one can build another set of test aggregates A_{test} from the same initial population. This test set can be acquired at a different time period and with different aggregated series than in the training set to simulate an attacker with knowledge about a different time period than the one he/she chooses to attack. The trained binary classifier is then tested on A_{test} and the accuracy is reported for each model.

The *MIA Risk* score is defined as the model accuracy score on all the aggregates of the test set: $risk$ = (True Positive + True Negative) / (True Positive + True Negative + False Positive + False Negative). A score of 1 means that the classifier performs perfectly (so the risk is maximum) while a score of 0.5 means that the classifier gives random predictions (because of the binary classification) and is not able to detect the target (i.e. the MIA failed). Note that other performance measures that put a strong emphasis on the aggregates that do contain the target series (i.e. the positive examples) could be considered as well (e.g. the F-measure, the true positive rate at a low false positive rate [3]) to define the MIA risk.

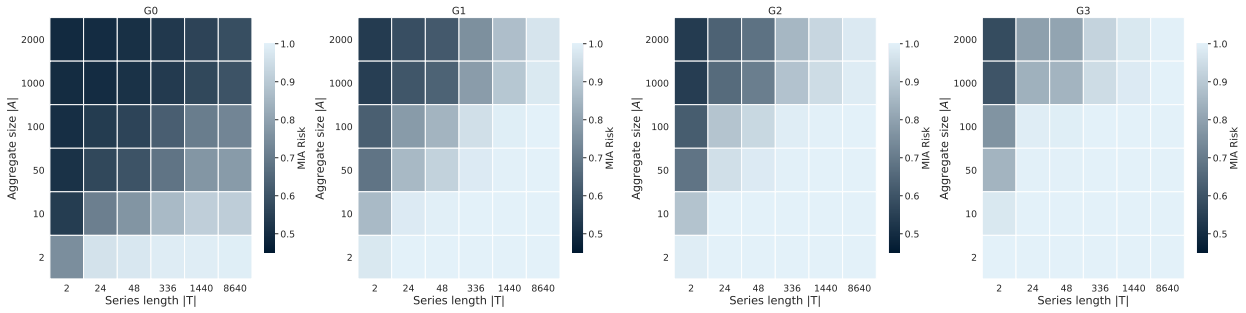
4.2 Choice of the classifier

Using a simple classifier (e.g. logistic regression [5]) would require handcrafting a number of features from the time series in order to give a fixed-size vector as input to the classifier (whatever the length of the series) and to tackle misaligned series. For instance, series from different time periods in the train/test sets could be misaligned. As handcrafted features, one can use traditional time series features such as its *mean, slope, min, max, var* and some spectral or wavelet transform features [1].

To be less dependent on the chosen features we have selected the recent, efficient, and very effective Minirocket [6] time series classifier. This classifier builds a fixed number of random convolutions that are then used as features by a linear classifier. Similarly to deep learning-based approaches, Minirocket automatically learns a good representation of the series (by means of the convolution kernels) that allows a non-temporal classifier (here, the linear classifier) to obtain excellent results for time series classification.

4.3 Experimental results on the ISSDA dataset

We use the half-hourly ISSDA dataset described in Section 2.2. To create our training/test sets, we have generated aggregates of size up to 2000 and of length up to 6 months



(a) Group G0 (b) Group G1 (c) Group G2 (d) Group G3

Figure 5: Average membership inference risks in function of the aggregate size, series length and oddness score.

(8640 timestamps). We study the impact of this size and length in the following.

The target individual time series were selected with different *oddness scores*. The *oddness score* is an estimation of the potential impact of a time series over the general population (i.e., the mean consumption). The higher this score for an individual series the further the series is from the population mean and, as shown below, the easier the MIA. The oddness score $\mathcal{O}$ over the set of series $\mathcal{S}$ and time period $\mathcal{T}$ is defined as follow: $\forall s \in \mathcal{S}, \mathcal{O}_s = \frac{\sqrt{\sum_{t \in \mathcal{T}} (avg(\mathcal{S})_t - \mathcal{S}_{s,t})^2}}{|\mathcal{T}|}$. We split the population according to the score distribution into four equal-size groups and select 10 target series from each group (i.e., in total, we learn/test 40 different classifiers for 40 different targets). The group "G0" contains series with the lowest oddness score while "G3" contains the strongest outliers which should be easier to attack.

The goal of our experiments is to measure how the membership inference risk depends on the aggregate size, the time series length, and the oddness score of the target.

Figure 5 summarizes the results obtained when the test set is designed from the same time period as the training set (but for different aggregates that do not overlap between the train/test) which corresponds to an attacker having a strong background knowledge. Lighter cells correspond to higher membership inference risk (averaged for all targets), i.e. a higher mean accuracy for all tested targets. As expected, publishing longer series leads to more successful MIA (the cells are lighter in the right parts of the sub-figures) since the classifier has access to more information to make a decision. Larger aggregate sizes are harder to attack since the impact of individual series is smoothed by the other members of the aggregate. The more distinct the target is, the easier it is to detect its presence inside an aggregate whatever the size and length of the aggregate: sub-figure d) which corresponds to G3 is overall much lighter than subfigure a) which corresponds to G0. Overall, publishing small aggregates over a "long" time period increases MIA risks. The oddness score of each individual should also be taken into account. All individuals in G3 are much more at risk than individuals in G0.

Figure 6 shows the MIA risk for fixed-length series (of 1440 timestamps) when the test set is designed with series from a different time period than the training set (one year after). As in Figure 5, the risk is directly correlated to the aggregate size and the oddness score of the target series. However, compared to the previous results on the same time period, we can see that the risk decreases quickly (but the predictions are better than a random guess) when the aggregate size is higher than 100 and stays low for all aggregate sizes in the G0 group (i.e. when the oddness score is the lowest). This shows that direct MIA is, unsurprisingly, less risky when the attacker does not have data from the target attacking period.

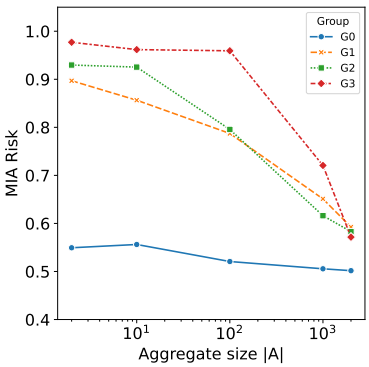


Figure 6: Average membership inference risks (accuracy) on the test set, function of the aggregate size and oddness score (G0 to G4). —T— = 1440 when training and test sets are collected from different time periods.

5. DATA RECONSTRUCTION RISKS

In this section, we address the problem of automatically detecting whether a publisher model is vulnerable to attacker models based on a formal approach in which publisher and attacker models are specified by queries. More precisely, a publisher model is expressed as *utility queries* specifying aggregate information that it seems useful to publish, while attacker models are expressed as *privacy queries* specifying data reconstruction attacks. In such an approach, a data reconstruction risk is formalized as the possibility of deriving an answer to a privacy query from some answers of utility queries. In [9], in order to deal with time series datasets, we have designed and implemented an algorithm that automatically detects all the data reconstruction risks raised by utility and privacy queries that are *temporal aggregate conjunctive queries*. In our framework, the utility queries are intended to be evaluated over private temporal knowledge graphs (which capture time series and their associated

meta-data in a uniform RDF data model) in order to build a public time series dataset (also in the form of a temporal knowledge graph).
The distinguishing point of our approach is to be *data-independent* and to come with an *explanation* based on the query expressions only. This explanation is intended to help data publishers to understand the data reconstruction risks for a given publisher model faced to a set of attacker models so that they can adapt their publisher model to mitigate the risks. Before summarizing our approach, we first illustrate it on the RDF version of the IRSSA dataset described in Section 2.2.

5.1 Illustration on the RDF ISSDA dataset

We assume that the publisher and attacker models are specified as queries over a common RDFS ontology [4].
Let us suppose that the publisher model specifies that it useful to publish:
(1) for each customer's number, their smart meter number;
(2) for each customer's number, their yearly income if it is more than 75000 and if they own their home;
(3) for each smart meter number, the sum of consumptions computed every hour over the measurement readings of the previous 3 hours.
This can be translated into the utility queries shown below by using SPARQL-like query language.

The utility queries expressing the publisher model

```
UQ1: SELECT ?sm ?o
        WHERE { ?sm issda:
            associatedOccupier ?o .
                 ?o issda:nbOfPersons ?n.
                    }
UQ2: SELECT ?o ?y
        WHERE { ?o issda:yearlyIncome ?y .
             ?o issda:own ?s. FILTER(?y >
            75000)}
UQ3: SELECT ?sm  ?timeWindowEnd SUM(?c)
          WHERE {(?sm issda:consumption ?c,
               ?ts)}.
          GROUP BY ?sm ?timeWindowEnd
          TIMEWINDOW (3h, 1h)
```

Now, suppose that the attacker model targets the following data reconstruction:
- the association between their smart meter number and their yearly income;
- their energy consumption measurements aggregated over intervals of 6 hours.
This can be translated into the following privacy queries for which no answer should be inferred from the published dataset.

The privacy queries expressing the attacker model

```
PQ1: SELECT ?sm ?y
        WHERE {?sm issda:
            associatedOccupier ?o .
          ?o issda:yearlyIncome ?y}
PQ2: SELECT  ?timeWindowEnd SUM(?c)
        WHERE {(?sm issda:consumption ?c ,
             ?ts)}
        GROUP BY  ?timeWindowEnd
        TIMEWINDOW (6h, 6h)
```

[4]available at `https://raw.githubusercontent.com/fr-anonymous/puck/main/issda\_schema.ttl`

With our approach, we can automatically detect several data reconstruction risks of the publisher model expressed by the above utility queries faced to an attacker model expressed by the above privacy queries, and provide the following explanations to the data publisher:
1) The first data reconstruction risk is due to the possibility of inferring an answer to PQ1 by combining answers to the utility queries UQ1 and UQ2.
2) The second data reconstruction risk is due to he possibility of inferring an answer to PQ2 from answers to the utility query UQ3 because:
a) PQ2 and UQ3 compute the same aggregate under the same conditions;
b) groups of UQ3 are partitions of groups of PQ2;
c) and finally, all time windows of PQ2 can be obtained as disjoint unions of some time windows of UQ3.
Based on the above explanations, the data publisher could modify his/her publisher model, for example by:
- removing one the utility queries UQ1 or UQ2;
- modifying the time window in UQ3, for instance by modifying the step between each consumption computation.

5.2 Algorithmic approach

The formal framework and the full characterization of privacy risks are described in [9]. Here we just summarize, and illustrate through examples, the principles underlying the verification algorithm: based on the query expressions only, it checks whether an answer of one the privacy queries can be inferred from answers to some utility queries.
In their most general form, the (privacy and utility) queries have 4 parts:
(i) a *graph pattern* that is an abstract specification (using a certain query language such as SPARQL) of the combinations between attributes/properties to be satisfied by the searched data ;
(ii) a set of *constraints* on the values of some of these attributes/properties to filter more precisely the searched data, using the FILTER constructor;
(iii) a *group by* part to specify the attributes/properties for which we want to group the searched data having the same values for those attributes/properties, using the GROUP BY constructor ;
(iv) a *result* defining the target attributes/properties the values of which must be returned by the query evaluation, and possibly aggregates to be computed on groups (specified in the *group by* part) using a given aggregate function.
When the aggregate function is computed on a dynamic property (such as *issda:consumption* in the ISSDA RDF dataset), *time windows* over which the aggregation must be computed must be specified. It is done using the TIMEWINDOW constructor with two parameters: a *size* to express the duration of each time window, and a *step* to express the time interval separating consecutive time window, which can thus be sliding (like in the UQ3 query of Section 5.1) or tumbling (like in the PQ2 query of Section 5.1).
The verification for a *simple* privacy query (i.e., without FILTER and GROUP-BY) against a set of any utility queries consists in checking whether the pattern of the privacy query is a sub-pattern of the union of patterns of some utility queries possibly joined by constraining some of their result attributes/properties to be equal. If this is the case, the corresponding utility queries are said *risky* for the privacy

query.
For example, up to variable renaming, the graph pattern of the SPARQL privacy query PQ_1:
?x1 issda:associatedOccupier ?x2 .
?x2 issda:yearlyIncome ?y2
is a sub-pattern of the pattern:
?x1 issda:associatedOccupier ?x2.
?x2 issda:nbOfPersons ?n.
?x2 issda:yearlyIncome ?y2
which is the joined union of the graph patterns of the two utility queries UQ_1 and UQ_2 obtained by equating the output variable $?y1$ of UQ_1 with the output variable $?x2$ of UQ_2.
This can be automatically detected, independently of the data. This exhibits a case where an answer to the privacy query PQ_1 can be derived from two answers to utility queries (for which the output variable $?y1$ of UQ_1 and the output variable $?x2$ of UQ_2 are instantiated with the same individual in the data).
For *complex* privacy queries, with FILTER and/or GROUP-BY constructors, we have to check in addition:

1. when the privacy query has a FILTER constructor, *whether* the FILTER constraints of the privacy query are compatible with the conjunction of FILTER constraints of the *risky* utility queries. This can be done using a CSP solver [5]

2. when the privacy query has a GROUP BY constructor, *whether* its graph pattern is isomophic (possibly up to a variable freezing) to the union of the graph patterns of the *risky* utility queries and its aggregate function is the same and applies to the same variable as at least one of the *risky* utility queries. This is the case for PQ2 and UQ3 in Section 5.1.

3. when , in addition, the privacy query has a TIMEWINDOW constructor, *whether* a time window for the privacy query can be obtained as the union of time windows of the *risky* utility queries when the aggregate function is MAX or MIN, or as the disjoint union of time windows of the *risky* utility queries when the aggregate function is SUM or COUNT [6]. This can be done using diophantine equation solver [7].

6. CHALLENGES FOR FUTURE WORK

In this paper, we have presented several approaches able to detect different types of privacy risks raised by publishing aggregates of (univariate and aligned) time series. We have highlighted some interpretable metrics (unicity, oddness) useful to measure the vulnerability to privacy risks of a time series dataset. We have also evaluated experimentally how the combination of some parameters of publisher and attacker models impact the risk. Finally, we have shown that machine learning approaches can be applied for predicting risk when formal methods or systematic simulation of attacks cannot be conducted. Depending on the methods used, we have indicated that some explanations can be provided to data publishers for helping them to understand and mitigate privacy risks. Here is a list of open challenges that should be considered:

- Evaluate the scalability of the methods to the length and the number of time series in real-world datasets (e.g. more than 35 Million time series for the French electrical provider, Enedis).

- Extend the models and the algorithms presented in this paper to more complex times series encountered in practice that may be multivariate and not necessarily aligned. The intrinsic complexity of systematic simulation approaches (such as the one presented in Section 3.2) is a limitation for their scalability. However, this is not such an important problem if they are used to build training datasets from which models can be automatically learned to predict privacy risks.

- Study the reliability of machine learning methods for modeling risk prediction on time series. Quantifying correctly the privacy risks using machine learning models requires machine learning methods with high accuracy. One challenge, outlined in Section 3.3, occurs when, by construction, the training dataset is highly imbalanced, thus making the accurate learning of the predictor very difficult for most of classifiers.

- For the approach described in Section 4 where we have seen that the learned model predicting membership inference risk performed worse when it is applied to a time period different from the one used in the training phase. It is a typical domain adaptation problem [23] that is made more complex because of the temporal aspect of the data and the multiple distribution shifts that can occur.

- Study how the use of machine learning techniques for predicting privacy risks can be used to construct *interpretable* explanations. A trade-off between prediction performance and interpretability should be achieved to obtain relevant explanation feedback with post-hoc explanation methods such as SHAP [13] or ANCHORS [19] applied to black-box prediction models.

- The deployment of formal methods, generalizing the one presented in Section 5, able to guarantee privacy by design based on a formal specification and automatic verification of publisher models compared to attacker models. This should allow to consider in particular attacker models corresponding to more abstract or less precise background knowledge about target users (e.g. holidays habits, heating habits, presence of a swimming pool, religion, etc.).

7. ACKNOWLEDGEMENT

This research was partially supported by TAILOR, a project funded by EU Horizon 2020 research and innovation program under GA No 952215.

[5]We used the python CSP library: https://pypi.org/project/CSP-Solver/
[6]We do not consider explicitly AVG because it can be computed by the union of 2 queries, one for computing SUM and the other one for computing COUNT.
[7]We used the Diophantine module of the python SymPy library : https://docs.sympy.org/latest/modules/solvers/diophantine.html

8. REFERENCES

[1] Anthony J. Bagnall, Jason Lines, Aaron Bostrom, James Large, and Eamonn J. Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data Min. Knowl. Discov.*, 31(3):606–660, 2017.

[2] Niklas Buescher, Spyros Boukoros, Stefan Bauregger, and Stefan Katzenbeisser. Two is not enough: Privacy assessment of aggregation schemes in smart metering. *Proceedings of Privacy Enhancing Technologies*, 2017.

[3] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, A. Terzis, and Florian Tramèr. Membership inference attacks from first principles. *2022 IEEE Symposium on Security and Privacy (SP)*, pages 1897–1914, 2021.

[4] Nitesh V. Chawla, Kevin W. Bowyer, Lawrence O. Hall, and W. Philip Kegelmeyer. Smote: Synthetic minority over-sampling technique. *J. Artif. Int. Res.*, 16(1), 2002.

[5] David R Cox. The regression analysis of binary sequences. *Journal of the Royal Statistical Society: Series B (Methodological)*, 1959.

[6] Angus Dempster, Daniel F Schmidt, and Geoffrey I Webb. Minirocket: A very fast (almost) deterministic transform for time series classification. In *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, pages 248–257, 2021.

[7] Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference (TCC)*, 2006.

[8] Flavio D Garcia and Bart Jacobs. Privacy-friendly energy-metering via homomorphic encryption. In *International Workshop on Security and Trust Management*, 2010.

[9] Christophe Bobineau Hira Asghar and Marie-Christine Rousset. Identifying Privacy Risks raised by Utility Queries. In *23rd International conference on Web Information Systems Engineering (WISE 2022)*, New York, United States, 2022. ACM.

[10] Klaus Kursawe, George Danezis, and Markulf Kohlweiss. Privacy-friendly aggregation for the smart-grid. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 175–191, 2011.

[11] Jack I Lerner and Deirdre K Mulligan. Taking the long view on the fourth amendment: Stored records and the sanctity of the home. *Stan. Tech. L. Rev.*, 2008.

[12] Rongxing Lu, Xiaohui Liang, Xu Li, Xiaodong Lin, and Xuemin Shen. Eppa: An efficient and privacy-preserving aggregation scheme for secure smart grid communications. *IEEE Transactions on Parallel and Distributed Systems*, 2012.

[13] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30, 2017.

[14] Francesca Naretto, Roberto Pellungrini, Anna Monreale, Franco Maria Nardini, and Mirco Musolesi. Predicting and explaining privacy risk exposure in mobility data. In Annalisa Appice, Grigorios Tsoumakas, Yannis Manolopoulos, and Stan Matwin, editors, *Discovery Science - 23rd International Conference, DS 2020, Thessaloniki, Greece, October 19-21, 2020, Proceedings*, volume 12323 of *Lecture Notes in Computer Science*, pages 403–418. Springer, 2020.

[15] Roberto Pellungrini, Luca Pappalardo, Francesca Pratesi, and Anna Monreale. A data mining approach to assess privacy risk in human mobility data. *CM Transactions on Intelligent Systems and Technology (TIST)*, 9(3), 2017.

[16] Francesca Pratesi, Anna Monreale, Roberto Trasarti, Fosca Giannotti, Dino Pedreschi, and Tadashi Yanagihara. PRUDEnce: a system for assessing Privacy Risk vs Utility in Data sharing Ecosystems. *Transactions on Data Privacy*, 11, 2018.

[17] Apostolos Pyrgelis. *Evaluating Privacy-Friendly Mobility Analytics on Aggregate Location Data*. PhD thesis, UCL (University College London), 2019.

[18] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro. Knock knock, who's there? membership inference on aggregate location data. In *25th Annual Network and Distributed System Security Symposium, NDSS*. The Internet Society, 2018.

[19] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In Sheila A. McIlraith and Kilian Q. Weinberger, editors, *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, pages 1527–1535. AAAI Press, 2018.

[20] Maria Rigaki and Sebastian Garcia. A survey of privacy attacks in machine learning. *ArXiv*, abs/2007.07646, 2020.

[21] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, 2017.

[22] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 3145–3153, 2017.

[23] Baochen Sun, Jiashi Feng, and Kate Saenko. Return of frustratingly easy domain adaptation. In *AAAI*, 2016.

[24] Latanya Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[25] Antonin Voyez, Tristan Allard, Gildas Avoine, Pierre Cauchois, Élisa Fromont, and Matthieu Simonin. Membership inference attacks on aggregated time series with linear programming. In Sabrina De Capitani

di Vimercati and Pierangela Samarati, editors, *Proceedings of the 19th International Conference on Security and Cryptography, SECRYPT*, pages 193–204. SCITEPRESS, 2022.

[26] Antonin Voyez, Tristan Allard, Gildas Avoine, Pierre Cauchois, Elisa Fromont, and Matthieu Simonin. Unique in the Smart Grid -The Privacy Cost of Fine-Grained Electrical Consumption Data. Preprint https://hal.archives-ouvertes.fr/hal-03833605, November 2022.

[27] Matthew D Zeiler. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.